ExtremeTech™

Christopher Negus ▪ Chuck Wolber

Linux® Toys

13 COOL PROJECTS FOR HOME, OFFICE AND ENTERTAINMENT

Build a home network server

Hello?

Set up a digital receptionist to receive voicemail in e-mail

Plug in, Turn on, Tune out

▫ Create a vintage arcade game player and a TV recorder/player

Linux® Toys

13 Cool Projects for Home, Office and Entertainment

Christopher Negus and Chuck Wolber

WILEY
Wiley Publishing, Inc.

Linux® Toys: 13 Cool Projects for Home, Office and Entertainment

Published by
Wiley Publishing, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Published simultaneously in Canada

Library of Congress Control Number: 2003101901

ISBN: 0-7645-2508-5

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

1O/ST/QZ/QT/IN

Library of Congress Control Number: 2003101901

About the Authors

Christopher Negus landed a job in 1984 at AT&T Bell Laboratories in Summit, New Jersey, in the group that developed the UNIX operating system. Because he had used UNIX before (and had even written programs in BASIC), he was considered "technical" and started out writing about computer network protocols.

Over the next eight years, Chris helped write (and rewrite) the thousands of pages of documentation that went with UNIX. He stayed with the same organization, even after AT&T spun it off into UNIX System Laboratories, then to Univel before selling the organization (with the UNIX source code) to Novell in 1992. (Yes, this is the infamous SCO source code!)

Over the next decade, Chris wrote or contributed to about a dozen UNIX books. In 1999, he made a transition to Linux with a vengeance when he wrote *Red Hat Linux Bible*. Since that time, the book has been a computer book bestseller and, in its various editions, has sold well over 175,000 copies worldwide.

As a follow-up to *Red Hat Linux Bible*, Chris wanted a project where he could take a more spirited approach to Linux and get more involved with the community aspects of open source software. Chris asked Chuck to put together some projects; they asked their friend Kevin Pedigo to review them, and they shared their experiences with members of their Linux Users Group. *Linux Toys* is the result of that process.

Chuck Wolber's first brush with computers came when he was 5 years old and got to bang away at a punch card machine with his Dad in the Purdue University computer lab. It was there that he learned that computers can be persnickety creatures and that it takes a careful hand to get them to do what you want. Since then he has worked on everything from the Commodore 16 and Apple II platforms to the big HP9000 machines that run much of the manufacturing at Boeing.

Since 1995, when he installed his first version of Slackware Linux, Chuck has been dedicated to the philosophy of open source software. With a 14.4K modem on his father's computer, Chuck downloaded that early Linux distribution one floppy disk at a time. When it installed flawlessly (the first time) and he saw his first `ls` command return the same result he was used to seeing on the big university computers, he was hooked.

Chuck is the founder and Chief Scientist of Quantum Linux Laboratories (`www.quantumlinux.com`), a consulting firm that offers a full range of open source software support for businesses. Through his company, Chuck provides remote and onsite systems administration, on-call support, remote backups, systems monitoring, security analysis, and intrusion detection.

Chuck lives in the Puget Sound area of Washington state with his partner, Karey; their fort-building/dragon-slaying son, Bailey; their dog, Heidi; Max the cat; and a fish that seems to be renamed at least once per week.

Credits

Acquisitions Editor
Debra Williams Cauley

Development Editor
Sara Shlaer

Production Editor
Eric Newman

Technical Editor
Kevin Pedigo

Copy Editor
C. M. Jones

Editorial Manager
Mary Beth Wakefield

Vice President & Executive Group Publisher
Richard Swadley

Vice President and Executive Publisher
Bob Ipsen

Vice President and Publisher
Joseph B. Wikert

Executive Editorial Director
Mary Bednarek

Project Coordinator
Maridee Ennis

Graphics and Production Specialists
Carrie Foster
Joyce Haughey
Lynsey Osborn
Heather Pope
Erin Zeltner

Quality Control Technicians
Laura Albert
John Tyler Connoley

Permissions Editor
Carmen Krikorian

Media Development Specialist
Travis Silvers

Proofreading and Indexing
Sossity R. Smith
Joan Griffitts

As always, I dedicate this book to my wife, Sheree.
—Chris

To Karey and Bailey
—Chuck

Preface

With free software and a few odds and ends, you can transform a cheap PC into a music or video archive box, a classic game machine, a home network server, or a handful of other projects. The only ingredients missing are the instructions to do it.

Enter *Linux Toys*.

Despite writing 1,000+ pages in *Red Hat Linux Bible*, I still get questions like, "How do I make a home network server?" or "How can I use Red Hat Linux as a dial-in server?" My answer is often something like, "Take a pinch of IP masquerading here, add DHCP there; then patch up your firewall like so and roll it into a bun."

Linux Toys is our attempt to bring together pieces of software and hardware to make some whole (and really cool) working projects. Because we're building them in Linux, the sky is the limit on where you can go with them. Add your own network connections, graphical interfaces, or back-end databases to a project, and you could grow our "Toys" into powerful home or office tools.

Where Did the Toys Come From?

Linux Toys came from all over the place. Well, all over the Internet, anyway.

When Chuck and I sat down to plan *Linux Toys*, we came up with a list of projects we thought could be done with Linux and free add-on software. We wanted each Linux Toys project to be entertaining or useful or fun or (preferably) all of the above.

Armed with our list, Chuck set out to find the free software that would suit our projects, and then he integrated those pieces. A few worked quite easily, while some others he glued together with Perl scripts or configuration settings. In one case, the string of patches Chuck sent into a project resulted in his taking over as the project's maintainer.

Every week or so, I'd sit on the floor of Chuck's office with his dog, Heidi, as Chuck showed me the latest Toys project. I'd take it home, try to re-create it on my computer, and then write it up in the chapters contained in this book. After that, our tech editor, Kevin Pedigo, would take the projects and see if he could get them working as I described.

Some of the projects are built purely from software that comes with Red Hat Linux. (Red Hat Linux is the basis for all our projects, except for DogHouse Linux in Chapter 13). Other projects are based almost completely on outside, stand-alone projects (such as VOCP in Chapter 10). Some require additional hardware, such as a TV card, a relay board, or a camera. Some will run on minimal hardware (DogHouse Linux) and some won't (anything video).

As we progressed, we found some of the projects to be really useful (we both have Linux Toys music players and digital picture frames running in our homes). Some projects were challenging to get running — in particular, video and voice modems are difficult. Other projects were just fun in a really dumb way (Look, my toy car can do a figure eight!). We hope that we've removed, or at least reduced, some of the more harrowing issues. When you are done, you should (we hope) end up with not only working projects but also a heightened imagination as to what you can do with Linux.

And remember: You can build more than just toys with this book. You can begin building a connection to a community of thousands of developers and millions of users around the world. To learn more, you can join a mailing list, read HOW-TOs, and even visit our own `LinuxToys.net` Web site to draw in more resources to help you on your journey with Linux.

How This Book Is Organized

There are six parts to this book. Of the numbered parts, Part I contains introductory material. Parts II, III, IV, and V contain the actual projects. The final section, consisting of the appendixes, contains information on getting and installing software, as well as a few basics on using Red Hat Linux. Here are expanded descriptions of the sections.

Part I: The Basics

Chapter 1 lays out the approach to the Linux Toys projects. Chapter 2 goes into detail about finding hardware and software.

Part II: Entertainment Projects

Chapters in this part contain sound, video, and game projects. Chapter 3 describes how to make a music CD ripper and player. Chapter 4 tells how to turn your home videos into digital files that you can store and play on your computer. Chapter 5 describes how to install and set up a television video recorder and player. Chapter 6 tells how to use MAME software to create a classic arcade game player.

Part III: Projects for the Home

These chapters contain useful home projects. Chapter 7 describes how to set up Red Hat Linux as a home network server, allowing other client computers to share an Internet connection, files, and printers. Chapter 8 describes how to broadcast streaming video, so you can watch your home from your computer at work or broadcast video to a couple of friends. Chapter 9 describes how to put together a temperature monitor and use it to monitor weather from your Linux system. Chapter 10 tells how to set up a voice modem in Linux that can be used to take telephone messages and mail those recorded messages to any e-mail box you choose.

Part IV: Small Business Opportunities

While intended more as exercises than as real business opportunities, the projects in this section describe the technology that Internet Service Providers and Web-hosting services use on

their servers. Chapter 11 tells how to set up Red Hat Linux as a dial-in server for a shared Internet connection, a DNS server, and an FTP server. Chapter 12 describes how to set up multiple virtual servers on a single computer, so friends and associates can have their own Web sites (complete with individual domain names).

Part V: Just for Fun

This part contains projects that are mostly just for fun. Chapter 13 describes how to make a DogHouse Linux distribution on floppy disk and then play a couple of classic, character-only games from the 1970s. Chapter 14 tells how to operate a remote control toy car using a relay board controlled from Linux. Chapter 15 tells how to convert an old laptop into a digital picture frame that displays your pictures on a rotating basis.

Note

All of the projects require a Pentium-class computer or better, except for the DogHouse Linux project. Once that project is copied to a floppy disk, it can be run on most PCs with a floppy drive (including 486 machines).

Appendixes

The appendixes contain supporting information for the rest of the book. Appendix A describes the Linux Toys software that is included with the book. Appendix B describes some of the basic Linux skills you need to use this book. Appendix C walks you through installing Red Hat Linux.

What You Need for the Projects

For all the projects, you need a PC, the three Red Hat Linux installation CDs, and this book. The book includes the Linux Toys CD, which has the software you need to add. The Red Hat Linux software can be obtained from *Red Hat Linux Bible*, directly from Red Hat, Inc., or other sources. Many of the projects also require some additional hardware and may place special demands on the PC itself.

Check Chapter 2 for an overview of the hardware and software requirements for Linux Toys. Then refer to each project chapter to determine the special requirements for each particular project.

The Linux Toys CD

The CD that comes with this book contains the software you need to complete the Linux Toys projects. Each chapter describes which packages from the CD you need for the project it describes. Most of the Linux Toys software is in RPM format (the format used to install software in Red Hat Linux).

Although Linux Toys software was built and tested for Red Hat Linux, we include the source code on the Linux Toys CD as well. If you are predisposed to do so, it should be possible to rebuild most projects for use with other Linux distributions.

The Linux Toys Web Sites

There are two Web sites associated with Linux Toys:

- **Linux Toys at Wiley** (`www.wiley.com/compbooks/negus`) — Wiley Publishing, the publisher of *Linux Toys*, maintains a Web page that pertains to issues surrounding the purchase and features of the book.
- **Linux Toys Community** (`www.linuxtoys.net`) — Come to the LinuxToys.net site to participate in the mailing list and generally share information about the Linux Toys projects.

Conventions Used in This Book

On occasion, I will want to highlight code or commands during a procedure. Here are some examples of text that is marked differently along the way.

Sometimes in a procedure, I want to make a distinction between what you type and what is returned. In those cases, the entire input and output is set in this `monospaced font`, and the part that you type is bold. For example:

```
# ssh toy
root@toy's password: *******
Last login: Thu May 22 12:58:49 2003 from music.linuxtoys.net
#
```

In this example, someone typed `ssh toy`, then typed a password (indicated by the asterisks in bold). The rest are the responses from the computer. This example shows a command typed to the shell. If you are new to Linux, you typically open a Terminal window to get to the shell. A prompt ending in a pound sign (#) means you should be the root user when you run the command; when you see a dollar sign ($), you can be any user.

Special icons for Notes, Chuck's Tech Tips, Cautions, and Cross-References will appear from time to time. Those paragraphs contain an extra bit of information, a special way of doing something, something to watch out for, or a pointer to another chapter, respectively. Here's an example:

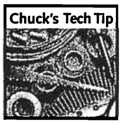
Note A Note contains an extra bit of information.

On occasion, I stop to expand on a feature that we happened to run into in the course of a procedure. In those cases, where I want to say more than a few words to expand on a topic, I'll put the information in a sidebar, where the title begins with "A Lesson in . . ." Here's what it looks like:

A Lesson in . . .

"Lessons" like these give you some guidance on a topic or skill you need for the project.

Throughout the book, we have interspersed technical tips from Chuck. We refer to these as Chuck's Tech Tips. Here's what a Chuck's Tech Tip looks like:



Chuck's Tech Tips provide technical explanations of some aspect of the topic at hand.

On with the Show

We hope that you are as excited to try out these projects as we are to bring them to you. If you are a Linux expert, feel free to jump right into the project of your choice. If you are new to Linux, be sure to go through the introductory materials and step through the appendixes to get a feel for how to use Linux.

Acknowledgments

Chris: I would like to acknowledge Linus Torvalds and the Linux kernel developers (`kernel.org`), Richard Stallman and the GNU Project (`gnu.org`), and Red Hat, Inc. (`redhat.com`), as major contributors to the Red Hat Linux distribution used as the foundation for projects in this book. Our book also relies heavily on software contributions from the Internet Software Consortium (`www.isc.org`), the Apache Software Foundation (`www.apache.org`), the Xfree86 Project (`www.xfree86.org`), and the GNOME Project (`www.gnome.org`).

These and other projects attest to what can be achieved through the open source development process.

I'd like to thank Debra Williams Cauley at Wiley for her strong leadership in keeping us on track to meet our publication dates and Sara Shlaer and Eric Newman for steering it through the development and production processes. Thanks to Margot Maley Hutchison at Waterside Productions for contracting the book with Wiley.

Special thanks to Chuck for spending countless hours researching, debugging, and scripting the projects in this book. Without his partnership on this book, I'm sure it would have been another few years in the making (if it ever got done at all).

Another special thanks to our technical editor, Kevin Pedigo, who has gone well beyond his job description, scrutinizing the projects and giving us insightful new directions in which to take them. Kevin's skill in building RPM packages and scripting some neat tricks for dealing with some audio and video problems was invaluable to us.

Thanks to Patrick Scanlon for taking all the awesome penguin images in the book. (Chuck and I are responsible for the less-awesome shots in the book.) In real life, I couldn't have afforded to hire him. So I was fortunate we were able to work out a buddy rate on the project.

On the personal side, thanks to my wife, Sheree, for being my sunshine during the rainy days in the Pacific Northwest. Thanks to Caleb and Seth for sticking it out and completing high school and preschool, respectively.

Chuck: I would like to thank the Linux kernel developers and the GNU Foundation for providing a platform that is steadfastly ruled by technical merit and openness rather than marketing deadlines and profit. Special thanks go to Linus Torvalds and Richard M. Stallman for their unwavering dedication to their principles.

I'd also like to thank all of the Linux application developers. Without applications, Linux goes nowhere. Your hard work and dedication are a large part of why this book was even possible. Many thanks go to SourceForge for providing the infrastructure on which a large portion of the application developers rely.

Everyone's hard work would be for naught if it weren't for the distribution integrators who "glue" the components together and make Linux easy to install. Thanks go to Red Hat, Inc., for providing the platform we used in this book. The simplicity of your design is why we use it.

We definitely couldn't have done this project without the help of our technical editor, Kevin Pedigo. At every turn he went above and beyond the call of duty to ensure that every detail was as correct as we could possibly make it.

Chris did nearly all of the writing for this book, which left me to do all of the fun stuff. Thanks for slaving away all those countless hours, and thanks for your patience when things didn't work quite right the first time (or the second or third or . . .).

The music files on the CD are from my good friend and business partner Bill Broomall. I often wonder, is there no limit to this man's talent?

On a personal note, I owe a debt of gratitude to my father. You can go many places in life, but nothing affects you like the experiences you have when you are young. I thank him for all the time he spent explaining complex subjects to me as well as for the many hours we spent working on projects together. He is a great example to me. I hope I can be such an example to my own son.

I've come to realize that one of the most difficult burdens of writing a book falls on family members of the author. Many thanks go to my partner, Karey Sayers, for her patience and encouragement while pulling way more than her fair share so I could work on these projects.

And above all, glory be to God!

Contents at a Glance

Contents

Part II: Entertainment Projects

Part III: Projects for the Home

The Basics

part

I

in this part

Exploring Linux Toys

chapter 1

in this chapter

- ☑ The spirit of Linux Toys
- ☑ The Linux Toys projects
- ☑ What is Linux and open source software?
- ☑ The Linux Toys CD-ROM
- ☑ The LinuxToys.net Web site

With a spare PC, the Linux operating system, a few added hardware pieces, and this book, anyone from a handy do-it-yourselfer to a Linux guru can end up with some fun and useful home or office projects. *Linux Toys* takes you from start to finish through the process of building cool devices for entertainment, home automation, small business, or just plain fun.

Although Linux Toys focuses on building projects, it's really much more than that. At its heart, Linux Toys is meant to reflect the spirit of the Linux community. Build a project and then find new ways to expand it, extend it, and configure it. Share your ideas with others and before you know it, your "free" software could become something more powerful than you can buy from any single vendor.

To get started, you don't need the most recent and hottest computer hardware. Some of the projects will run great on that eight-year-old PC sitting in your closet (such as the Linux Home Network server in Chapter 7 or Web Hosting Service in Chapter 12). Other projects will do better by adding a larger hard disk, some RAM, or a new video card.

If you already have Red Hat Linux running on your hot new computer, you can pop in the software from the Linux Toys CD to get the best possible performance for the various video and audio projects (see Chapters 3, 4, 5, and 8). You'll rely on a combination of software that comes with Red Hat Linux and some neat software projects we dug up from the open source software community.

While the spirit of this book is one of fun and community, the technology we describe is quite serious and becoming more powerful each day. Some of the same software we describe is running the server computers for companies around the world. We just choose to not be stuffy in how we go about using it.

Taking the First Step . . .

If the possibilities mentioned previously sound cool to you, we welcome you to *Linux Toys*. Your guides to the world of Linux Toys are Chris Negus (who wrote most of the words you see here) and Chuck Wolber (who built most of the projects).

Like Linux itself, the book was created in a spirit of fun and freedom, while still resulting in some serious, useful projects. *Linux Toys* acts as a "cook book" to step you through the process of building a number of projects. For each project, we list the hardware and software required. Then we go through installing, setting up, and running it. Finally, we give tips for enhancing the project or taking it in an all-new direction.

Has your computer got what it takes?

For each project, we lay out the requirements for the computer you need and tell you if you need some extra parts (such as a TV capture card or a modem). In Chapter 2, we step you through ways of evaluating how suitable a PC is for the projects.

In most cases, the minimum Pentium-class PC needed to install Red Hat Linux will work fine for our projects. When bigger iron is needed, we'll let you know. Unlike with some of the latest commercial operating systems, however, you won't need the hottest new PC to get into Linux Toys.

A lot of the fun of these projects can be to breathe new life into an old, ignored PC. For example, with the digital picture frame project (Chapter 15), we encourage you to hunt down a $25 laptop from a local surplus store and mount it in a real picture frame to display your digital images.

We'll do our best to guide you through the process. Think of us as you would Norm and Tom, walking you around a project in *This Old House*, or Rachel Ashwell, shopping for flea-market bargains in *Shabby Chic* television shows. Our own "This Old Computer" and "Shabby Geek" approach will help you sort through and put together the parts you need to get your projects up and running.

After you've done a project, we want you to come out with:

- Something useful (or at least fun)
- Something you can expand on your own as your spirit takes you
- Knowledge of what Linux is and ways to use it
- New friends at `Linuxtoys.net` to share ideas and enhancements with (more on this later)

If everything works out the way we hope, your new project will fit seamlessly into the lifestyle to which you have become accustomed (see Figure 1-1).

Have you got what it takes?

When you hear that someone has set up his or her stereo, home security system, and toaster so that it can be operated by a single remote control, do you "Oooh and ahhh" out loud? When

you and your friends start talking about your latest home projects, do your spouses discreetly move to the other side of the room? If so, then you are our people!

As for your expertise, I expect you to be computer literate, but not necessarily an expert in Linux. For example, if I were to point to a PC's mouse, keyboard, video card, serial port, parallel port, and monitor, you should be able to pipe up with what each of those things are. If I ask you if you know what text editors, application programs, and Web browsers are, you should nod at me knowingly. Beyond that level of information, you should be able to follow along (with the help of some Linux basics at the back of this book).

If you are a genius and our toys have set off a maelstrom in your brain of ways to improve on or take a new direction with any of our projects, we welcome you with open arms. Our projects are built on open source software and put back into the community as open source software. Read the GNU General Public License (described later); then extend, expand, and enhance to your heart's content. We have made the `LinuxToys.net` site as a gathering place for you and others who want to go beyond the boundaries of this book.

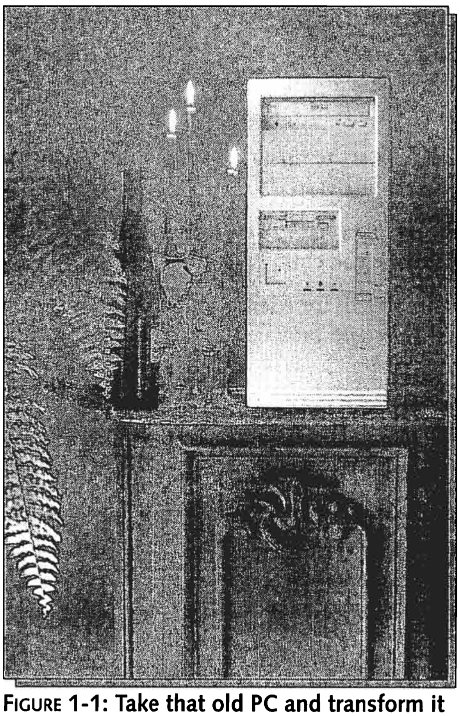


FIGURE 1-1: Take that old PC and transform it into a Linux toy that fits your lifestyle.

Checking Out the Projects

Linux Toys projects range from simple music and video players, to useful home and small-business tools, to some lighter projects that are just flat-out fun. The 13 Linux Toys projects include

- **Music jukebox** — Automatically rip and compress music CDs to your hard disk; then play them continuously at random or from a playlist you create. Music is stored by artist name, CD title, and track names.
- **Home video archive** — Make backup copies of your precious home videos, compress them, and burn them to CD or DVD.
- **TV recorder/player** — Set your Linux PC to automatically record and play back your favorite television shows by selecting shows from a Web-based interface.
- **Arcade game player** — Use the xmame game emulator to play old console games.
- **Home network** — Set up a network to share an Internet connection, do file and printer sharing, and act as a routing firewall (to protect from intruders on the Internet).
- **Home broadcast center** — Send streaming video to other computers on your network or the Internet. Use the project to watch home-surveillance cameras, do a mini video conference, or share broadcast television around your home.
- **Temperature monitor** — Gather, store, and chart temperatures; then add the output to a Web page, e-mail signature, or graph.
- **Telephone answering center** — Set up multiple voice-mail boxes in Linux so that audio telephone messages can be e-mailed to anyone you choose.
- **Internet Service Provider** — Become a small ISP for friends and family by using a Linux server to connect dial-in computer lines and provide support for multiple users for Web, FTP, and DNS service.
- **Web-hosting service** — Set up a virtual hosting DNS server, so the server can provide Web content form multiple domain names and allow your clients to add their own Web content.
- **Linux on a Floppy and BSD Games** — Play with some Linux commands and classic text-based games from a bootable floppy disk.
- **Remote control car** — Control a remote control toy car from your Linux system.
- **Digital picture frame** — Have your digital pictures display randomly from laptop components installed in a picture frame.

Along with these projects are general information sections that will help support other projects. These include:

- **Using Linux** — Get the basics you need to know to use the Linux operating system, in particular Red Hat Linux. (See Appendix B.)
- **Installing Linux** — Learn to install Red Hat Linux and get started using it.

Some of these projects are designed to run as single-use devices. Some of the reasons for using only one project per machine are

- **Some projects take over your computer.** For example, the music jukebox assumes that every CD you put in is meant to be copied to the hard disk and takes over your sound card by playing music continuously.
- **Minimal hardware requirements.** To keep down the use of resources, we tried specifically to note the minimal software and hardware you need for each project. Different projects have different minimums.

Just because some projects can work well as single-use devices on less powerful PCs, don't let that limit you. Many of the projects will co-exist quite nicely on the same PC. Others can be turned off and on so as not to conflict with each other. Also, if you have a new, spiffed up PC you can certainly use that to build these projects. It's just tough for us to make a case, cost-wise, to use a new PC to operate a toy car.

This brings us to the question of costs.

Costs of building Linux Toys

Will it cost you a lot to build these projects? The short answer is "No, it doesn't have to." The longer answer is, "Have you ever heard the old story about stone soup?"

In one incarnation of the story, after an old woman refuses to give a hungry tramp some food, the tramp insists that he can make them both some wonderful soup from nothing but a stone and some water. In the process, he gets the woman to add a bit of flour, some carrots, a few potatoes, and so on until the cupboards are empty and the soup is, well, state of the art. And the woman, of course, is thrilled to have such a fine soup from nothing but a stone and water.

We believe that these Linux Toys projects will each bear something greater than the stone and water you put into them. If, however, you decide to empty your cupboards to add an expensive stereo on the back end of our jukebox or buy a 120GB hard disk to store your home videos, that is entirely up to you. Go for it.

Our goal at the onset is to make each project worth the price of the book in what it adds to your old PC. If it turned out that a project only seemed worth about $19.95, we tried to add $9.05 worth of enhancement suggestions and links to related projects that might interest you.

Finding hardware

Most of the projects can start with an inexpensive PC at their base. However, not just any PC will be appropriate for every project. Some might require larger hard disks, more RAM, or a better processor than the PC you stumble upon. We'll give you some guidelines for finding the hardware that will work for these projects.

For some of the trickier pieces of hardware, such as voice modems or TV capture cards, we recommend exact models that worked well for us. Other hardware you might be able to find in a used computer store or even in your own basement (see Figure 1-2).

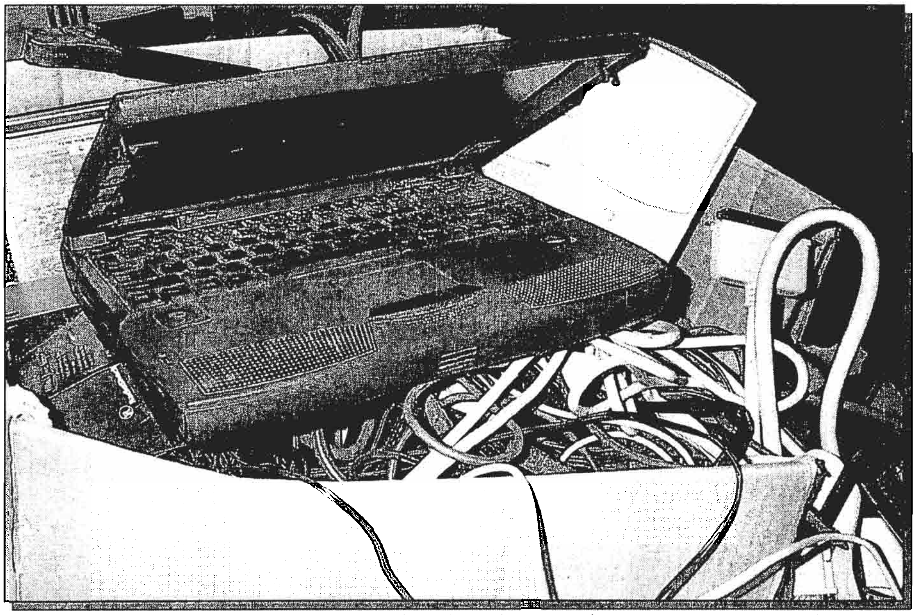


FIGURE 1-2: An old laptop could make a fine digital picture frame or toy car controller.

Getting Started with Linux Toys

The basic recipe for our projects is:

```
PC + Linux + Linux Toys CD + (bits of hardware) = Cool Projects!
```

I have an old circa-1994 computer that would need some enhancements to work effectively for some of the projects. For example, my 1GB hard drive (which was large at the time) won't hold the full music jukebox as we describe it. It will, however, hold a minimal Red Hat Linux installation and run a temperature monitor quite nicely.

Each project lists the hardware and software you need, so you can be sure to have them ready before you begin. In fact, you might want to go through those lists before you buy the book.

The ingredients for our Linux Toys recipes are described in the following paragraphs.

The PC

Because we are using Red Hat Linux, our minimum PC requirements are set by Red Hat's minimum. For that reason, a Pentium or compatible processor is the least powerful CPU allowed for our projects. So, we don't support these projects on 386 or 486 processors. (The one exception is the Linux on a floppy disk project described in Chapter 13, which will run on most computers that have a floppy disk drive.)

Earlier versions of Red Hat Linux did support 486 processors. If you want to take the source code of our projects and compile them on an earlier Red Hat Linux distribution (or some other Linux distribution), give it a try. Let us know if it works and we'll let others know.

Despite the fact that we don't support pre-Pentium PCs, you can find plenty of usable computers for only a few dollars . . . if you know where and how to look. We'll help you with some tips in Chapter 2 for shopping for Linux Toys hardware.

The operating system: Red Hat Linux

Linux is a free operating system based on an operating system called UNIX and includes software created by thousands of programmers around the world. There are different distributions of Linux, from which we have chosen to use Red Hat Linux.

In particular, we built Linux Toys to run on Red Hat Linux 9 and the following release. However, we also tested many of the Linux Toys packages on Red Hat versions 7.3 and 8 and found that they worked well.

If you are not familiar with Linux (or not exactly sure what an operating system is), here are a few questions and answers to help you.

- **What is an operating system?** An operating system is software that lets your computer software talk to your computer hardware. It can:
  - Manage access to the computer's processor by the programs you run.
  - Provide a structure to the files and directories that contain your programs and data.
  - Include drivers for different hardware devices, so your programs can talk to your CD drive, sound card, Webcam, or other computer hardware (without having to be burdened with the details of each hardware device).

  Advanced operating systems such as Linux were also designed to manage multiple users, multiple running programs (called processes), and multiple processors (CPUs) at the same time.
- **What are other examples of operating systems?** Microsoft Windows (95, 98, 2000, NT, ME, and XP) and Mac OS are other operating systems.
- **Do I need a commercial operating system, such as Microsoft Windows, on my PC?** You don't. No commercial software is needed with our projects; you can erase your hard disk and use only free software. In fact, Red Hat goes to great lengths to not bundle in any software (even some software claiming to be free) that may infringe on legal patents or limit redistribution of the operating system. We try to do the same.
- **What is Linux?** Linux is an open source operating system. Ways in which you can copy, distribute, and modify Linux are covered under terms in the GNU General Public License (GPL). At the heart of the GPL is the intention that if you use code covered by GPL, you can't restrict others from using that code. In other words, add to it and pass it on.

In its purest form, Linux is really just the core of an operating system, referred to as the "kernel." To create the Linux kernel, in 1991 Linus Torvalds used some code that already existed (mostly GPL software) and some code he wrote himself, then put it out on the Internet for other programmers to enhance and improve. They did.

Within the Linux kernel, which Linus Torvalds continues to maintain, are mostly things that you don't see. Those things include hardware drivers (that let applications communicate with different computer hardware) and file system types (to store and organize files and directories).

Although there are other open source operating systems, Linux has grabbed the lead as the premiere operating system platform for running GPL software.

Chuck's Tech Tip

In the beginning, DARPA (Defense Advanced Research Projects Agency) saw the need for an efficient and secure multi-user operating system. From this vision, the MULTICS project was born. The project itself never saw completion. However, engineers from Bell Laboratories, one of the project's contractors, saw some real potential from what was learned and continued developing it into what they called UNICS. UNICS, a singular version of the original MULTICS, was later shortened to UNIX. Nearly 30 years later Linus Torvalds made his implementation of UNIX available to the world. It was a friend of his who, to Linus's initial embarrassment, suggested the name Linux.

- **Why are there different Linux distributions?** Think of Linux as a car engine. To use an engine, you need to add a car body, wheels, seats, and controls (like steering wheels and gas pedals). Because Linux is under GPL, companies and organizations are free to take it and add the pieces needed to make a complete operating system. Each adds its own "parts" to the Linux kernel, such as a graphical user interface, tools for administering Linux, and software development tools.

  Some of the more popular Linux distributions include Debian, SUSE, Mandrake, Gentoo, Lycoris, TurboLinux, Yellow Dog, and Slackware. Heck, you can even build your own Linux distribution if you like. What makes them the same is that they include the same Linux kernel and many of the same open source software projects (such as KDE and GNOME desktops, Apache Web server, and Samba file/print sharing). What makes them different are things such as installation procedures, administration tools, and which of the thousands of available open source software components they do or don't include.

- **What is Red Hat Linux?** Red Hat Linux is the most popular commercial Linux distribution. Red Hat Inc. has some powerful partners, such as IBM, Hewlett-Packard, and Dell that offer Red Hat Linux on everything from desktop PCs to large multi-processor servers.

  Personally, Chuck and I like Red Hat Linux for a few reasons. We have found it to be a reliable and stable operating system for the server and desktop systems we use in our work. Also, Red Hat has worked very hard to keep its operating system as pure as possible within the letter and spirit of the free software movement.

- **Can we use a different Linux distribution for Linux Toys?** The answers are no or maybe. The binary RPMs (binary software packages) on this book's CD-ROM expect to be installed on a Red Hat Linux system. Although you can surely adapt any of these

projects to other Linux distributions, there will be enough differences in our step-by-step procedures that a new Linux user might stumble a few times. That's the "no" part.

For the "maybe" part, remember that we publish the source code. If you are a programmer, there is not too much in these projects that, with a small amount of modification, you couldn't adapt to a different Linux distribution. You are free to do so.

- **How do I get Red Hat Linux?** There are a lot of ways to get Red Hat Linux. If you are serious about learning Red Hat Linux, you can purchase *Red Hat Linux Bible* (written by Christopher Negus and published by Wiley), which includes the complete Red Hat Linux distribution on three CDs. I have tried to make the book a comprehensive tool for users and administrators. It weighs in at over 1,100 pages.

  If you have a high-speed Internet connection (or a slow connection and a lot of patience) you can download Red Hat Linux from various FTP sites on the Internet. There is no charge for downloading Red Hat Linux from the Internet.

The Linux Toys CD-ROM

The Linux Toys CD-ROM contains the software we added to what already comes with Red Hat Linux to make the projects go. If you are a novice Linux user, you can follow the instructions for installing the RPMs (binary software packages) for each project. More advanced users can compile the software themselves from source code provided at the `LinuxToys.net` Web site.

The software on the CD is a combination of:

- **Open source projects** — Instead of starting from scratch, when we had an idea for a Linux Toys project, we searched the open source community to see if others had begun similar projects. We compiled and, in some cases, modified code and chose configuration options for these software projects to build our Linux Toys projects.
- **Glue** — To bring the various software components together into a cohesive project, Chuck integrated the software and created such things as start-up scripts and related software. The goal was to have this code "glue" together the software from different projects to create projects that would just start and run as simply as possible.

Because the Linux Toys CD contains only open source code, you are free to copy and redistribute its contents under the rules of the GPL.

Odd pieces of hardware

For many of the projects, you will need to get an extra piece of hardware or two for them to work. In most cases, the basic set-ups require very little extra hardware.

An example of a piece of hardware that could end up costing more than your used PC is the LynX-PORT board. Although the board is expensive for running a remote control car, it could be reused for other home automation projects. We also feel this particular board is cool enough that you will find other good uses for it, once we show you how to use it from Linux.

Using Open Source Software

In recent years, the term "free software" has generally been replaced by the term "open source software" in hopes to clarify what the movement is all about. Just as "freedom" in society doesn't mean you can do anything you want (such as restrict other people's freedom), freedom with open source software comes with some responsibility. The responsibilities are meant to encourage continued development of free software, when you use the code in certain ways. (See the "Open Source Software Definition" sidebar.)

When a programmer creates open source software, he or she typically attaches one of many available open source licenses to it, defining how it can be used. The intention of most open source licenses is to encourage people to make changes to the software and share those changes with others.

Open Source Software Definition

The Open Source Definition, written by Bruce Perens, sets down nine points defining open source software (see `www.opensource.org/docs/definition_plain.html` for the complete definition). Here are the points, with my interpretations of what they mean to you as someone creating or using open source software:

- **Free Redistribution**—The software creator can't keep you from selling or giving away the software as part of your software project and they can't make you pay a fee for it.
- **Source Code**—The software creator must give you source code or make it available.
- **Derived Works**—The software creator must let you redistribute the software, with your changes, under the same license.
- **Integrity of the Author's Source Code**—If you modify the software, the software creator can ask you to change the name or version, to protect the original code's integrity.
- **No Discrimination against Persons or Groups**—The software creator can't say, "XYZ people can't use my software because I don't like them."
- **No Discrimination against Fields of Endeavor**—The software creator can't say, "This software can't be used to study extraterrestrials or create recipes for moonshine."
- **Distribution of License**—Everyone who uses the software can use it under the same license, without needing to add a license.
- **License Must Not Be Specific to a Product**—The software creator can't restrict use of the code to a particular operating system or other software.
- **The License Must Not Restrict Other Software**—The software creator can't restrict you from distributing other software with the open source software. (The example given is that the license can't say that the software can be distributed only with other open source software.)

Understanding GPL and other licences

The most popular of the open source licenses (and the one that covers most software in Red Hat Linux) is the GNU General Public License (GPL) from the Free Software Foundation (`www.gnu.org`). You can get that license here: `www.gnu.org/licenses/gpl.html`. It's also printed in the back of this book and is on the CD. As the GPL states, when software authors commit to making their software GPL, the license allows you to:

- Distribute copies of free software (and even charge for this service if you wish)
- Receive source code or can get it easily if you want it
- Change the software or use pieces of it in new free programs
- Know you can do these things

There are lots of other licenses that can be used to cover open source-ish kinds of software. The Free Software Foundation maintains a list of some of these licenses (with their own comments about them) at their Web site (`www.gnu.org/licenses/license-list.html`).

Building projects with open source

As for using Linux and other open source software for building useful and expandable projects, there are several reasons why we think open source software is the best way to go:

- **No licensing fee for each toy.** You can build one or 1,000 toys from your Red Hat Linux distribution, and, other than the time you spend, it won't cost you any more than the nothing you paid in the first place (or just the one copy of Red Hat Linux you bought).
- **No hiding the code.** You can see and change all the code in your Linux Toys. If you don't like what it does, rewrite it yourself.
- **Get enhancements going forward.** The open source projects that make up Red Hat Linux and Linux Toys will continue to go forward and offer enhancements to their projects. We also hope people will make direct contributions to improving the Linux Toys projects. To you, it means that automatic improvements to the Linux Toys projects will be available to you. (Did I mention, at no cost?)
- **You can learn Linux.** Every Linux Toys project can benefit from the fact that you are building it on a full-service operating system. Learn Linux features for configuring a network connection, a cron job (to launch the project at set times), a TV capture card, or a Web interface, and you have multiplied the power of your Linux Toys project. At the same time, you'll be learning a powerful, professional-quality operating system.

Remember that open sources licenses are built on commitments to freedom and community. There are those who would equate people who create or use open source software with thieves and scoundrels. In my experience from Linux mailing lists and my local Linux User Group (LUG), I have found open source devotees as a group to be:

- **The first ones to help you.** Open source supporters always lend a hand if you have a problem with or question about Linux.
- **Respectful of copywrited material.** Open source supporters believe that CDs, DVDs, books, and software should be obtained legally, with proper compensation given to the works' originators. However, they also believe that people should be able to play their legally obtained music and movies on the players they choose (including their PC-based Linux systems).
- **Interested in making things work.** There is a committment in this community to getting your Linux box to work with any application or computer on your network. Compare that to a company that has a vested interest in selling you more of their products and fewer of the competitor's products. Sometimes they go out of their way to break software that tries to interoperate with their products.

The bottom line is that there are people around who will try to help you overcome obstacles you run into with Linux. Get on a Linux newsgroup or mailing list. Check out the resources at `LinuxToys.net`. If you can get on the Internet, there's help out there.

Chuck's Tech Tip

Many newcomers to the open source world find themselves disappointed or turned off by some of the responses they receive to their queries for help. Although every group has some bad apples, open source developers are generally very happy to provide help to end users. It is important to remember, however, that developers are very busy people and that requests for help take them away from developing and improving their projects. The best way to get good help is to do your homework first. Read as much documentation as possible; you might find that you can solve your own problem. If you still need help, don't be afraid to search the Web to see if anyone has asked the question before you. If you still need help, be as detailed in your query as possible.

Connecting with Linux User Groups

If you want to meet other Linux enthusiasts in your area, search out a local Linux User Group (LUG). Many of these groups offer monthly meetings, mailing lists, and Web sites to support their users. To search a list of LUGs for one that is near you, go to the Linux Online User Groups page (`www.linux.org/groups/index.html`).

Visiting LinuxToys.net

When the `LinuxToys.net` domain became available a short time ago, I snapped it up. My vision was to take the projects we are building and set up `LinuxToys.net` to be the focal point for continuing development and community support. Figure 1-3 shows the `LinuxToys.net` Web site.

FIGURE 1-3: Visit LinuxToys.net for the latest project information.

You need to visit `LinuxToys.net` to see its current features, since we intend the site itself to evolve. However, you can expect to find at least the following features when you visit `LinuxToys.net`:

- **Links to Linux Toys project sites:** For each project, you can follow links to its project site. From there, you can download the source code, check for available patches, and find any contributions that have been made to the project.
- **Mailing list:** You can subscribe to the Linux Toys mailing list to share your experiences and ask questions about the projects.

By publishing the source code and encouraging community interaction at `LinuxToys.net`, we are trying to remain true to the spirit of open source software. Just as Red Hat Inc. does with Red Hat Linux, we want to keep our projects free from patented material. For example, we use Ogg Vorbis to compress music for our music jukebox. Likewise, we want to discourage you from illegally sharing commercial software or using copy-protected music you don't own.

Going Forward

If you are familiar with Linux and are already a hardware guru, you can look at Chapter 2 ("Finding Hardware and Software") and proceed right to whatever project you want to do first. If you have no experience with Linux, you should probably do a bit more groundwork.

Saying that you can do these projects without knowing any Linux is a bit of an overstatement. It is a lot like saying you can go to France and speak with the natives using a book of popular French phrases. It will help you find the bathroom and order roast beef, but it won't get you much further.

The appendixes at the end of this book are meant to expose you to the basics of installing and using Red Hat Linux. As someone new to Linux, you should at least read Chapter 2 and check the appendixes (for Linux essentials) before you start a project. If you are serious about learning Linux, however, I strongly recommend getting *Red Hat Linux Bible* or another book that tries to thoroughly cover Red Hat Linux. It will serve you well over time, especially as you look to expand beyond the basic projects in this book.

Summary

Linux Toys was created to help you build neat projects with a PC and open source software, as well as to encourage you to learn more about Linux and become engaged in the open source community. The projects offered include entertainment, home, small business, and fun projects. You should come away from this chapter understanding a bit more about Linux, open source software, and what you need to move forward with Linux Toys.

Finding Hardware and Software

chapter 2

in this chapter

- ☑ Choosing a PC for projects
- ☑ Using kernel hardware information
- ☑ Finding open source software
- ☑ Hooking up with mailing lists and LUGs

Not every piece of hardware that you can plug a keyboard and mouse into will run Linux Toys. Some brand-spanking new PCs may not have drivers supported in Linux yet. Some old hardware may no longer be supported by the latest Linux kernel.

If the PC in front of you will work, that's great. If not, you may need to do some hunting around. So, to help you get your hands on a PC that will work for the projects in this book, we're going to give you some tips on:

- Evaluating the PC you've got in front of you
- Finding a PC that will work for you (if yours does not)

Our search entails taking you on a shopping trip. But instead of coming home with that perfect little love seat for your den, we're going to help you find the stuff that Linux Toys are made of: computer hardware and software.

Because the software needed for Linux Toys is free, "shopping" for software entails finding open source projects and finding people who build, use, and chat about the software. You shouldn't *need* any software besides Red Hat Linux and the Linux Toys CD that comes with this book. But you may find you want other software to extend your projects.

To begin with, let's check out that PC sitting in front of you.

Will This Old PC Work for Linux Toys?

You want to start building your Linux Toys and you find an old PC in your closet or at a garage sale or a new PC with some spare disk space. How do you know if the PC will work for any of the projects? Your choices at this point are:

- **Evaluate the PC** — If you'd like to know what you have (or if Red Hat Linux installation fails), you can evaluate what's in that PC.
- **Install Red Hat Linux** — If you don't care about the data on the computer, you can just try installing Red Hat Linux. It may just work, so why not? (See Appendix C for information on installing Red Hat Linux.)

In most cases (especially if you are new to Linux) you should go the "evaluate the PC" route. The reason is that you will inevitably run into some problem at some point. So when you go to a mailing list or support organization to ask a question and they ask about your hardware, you don't have to just say, "Uhhh, I dunno."

Likewise, if someone is selling you a PC and lets you try it, there is a lot you can do to find out about the health and usability of the computer.

Does the PC work?

Before you purchase that bargain PC for building your Linux Toys, you might want to make sure that it is working. Here are some quick steps to evaluate what that old PC has going for it.

Booting the computer

Begin by booting the computer and checking that it is able to start up.

1. Plug in the computer and turn it on (popping out any floppy disk or CD).
2. Watch as the BIOS starts up (see Appendix C for information about what happens when the BIOS starts) and look for something like the following:

   ```
   To enter Setup, press <f2>
   ```

3. Press the key that it says to enter setup (probably F2). You should see a screen that shows some basic information about your computer. Look for the following:
   - Processor type
   - Processor speed
   - Total memory

   Note this information so that if the computer boots at all, you can determine if it has the power the system needs for the projects you want to build. You might also see information about the computer's motherboard at this time.

Note Minimum processor type, processor speed, and memory requirements for Red Hat Linux are described in Appendix C. Where to find more complete information on supported hardware is described in the "Using Kernel Hardware Information" section in this chapter.

4. Exit the setup (being careful not to change anything unless you know what you are doing!) to start the boot process. At this point, the computer will either:
   - Start the operating system (Windows, Linux, or whatever) or
   - Run the boot loader (in which case, if there are multiple operating systems on the computer, you should choose the one you want to boot)

If the operating system starts properly, you can do a lot more to check the value of the computer. A failure to boot may be caused by a bad hard disk (or it may not). Sometimes auctions and computer stores erase the hard disk to avoid software-licensing issues.

Tip

A friend of ours who works at an auction actually puts Linux on used PCs to avoid licensing issues. If there is no OS on the computer you want, why not ask if you can install Linux on it before you buy it?

Checking for broken hardware

After you have tried to turn on the PC, there are a few components you can check that might not be working on a used computer:

- **Fans** — Check that all the fans are running. There is a fan on the CPU heatsink and one or more on the chassis. A fan that isn't working might have caused damage to other parts because of overheating. (The heatsink is a small piece consisting of little metal partitions that allow heat to dissipate from the CPU.)
- **Power supplies** — If the PC won't turn on at all, it may have a burned out power supply. That PC might be a bargain (who knows, it isn't running). If you buy a computer with a burned out power supply, it will cost about $20 to get a new 250W ATX power supply just to find out what you have. That's fairly cheap if you end up with a nice motherboard, but again, you won't know until it boots.
- **Hard disks** — A computer that begins the boot process but doesn't start an operating system might have a bad hard disk. Hard disk prices have dropped dramatically in recent years, so used ones can be had for just a few dollars. Because you can pay about $60 (or less) for a new 20GB hard disk, you can imagine what a used 5GB hard disk might cost you if you just want to build a toy. If the data is critical on the computer, however, I recommend you either buy a new hard disk or do frequent backups.
- **Monitors** — With new 15" CRT (cathode ray tube) monitors beginning at about $100, monitors that don't work often cost too much to repair. Used 14"–15" monitors that work can be had starting at about $10. While "burn-in" is not as much a problem as it was in the old days, CRT screens can dim over time. Also, power supplies can burn out.

The quality of the components in your computer can make a big difference in its value. If the operating system does start up, go through the following procedures for evaluating the hardware you have.

Does the PC have what I need?

If the PC seems to be working, you want to check that:

- The type of hardware contained in the PC is supported in Linux
- The PC contains the minimum components needed to run Red Hat Linux

Later in this chapter, I step through some of the requirements for each component of your PC. Appendix C describes the minimum hardware requirements for running Red Hat Linux (which forms the foundation of our projects). Also, the chapter for each project lists its own special hardware needs.

Evaluating hardware in Linux

There are both command-line and GUI tools for determining the hardware on your PC in Linux. From the GUI, try the Hardware Browser window; from the command line, start with the `lspci` command.

Running the Hardware Browser

If you are using a GUI, you can display basic hardware information about your computer using the Hardware Browser window. To open that window, click the Red Hat Menu ➔ System Tools ➔ Hardware Browser. Figure 2-1 shows an example of that window.

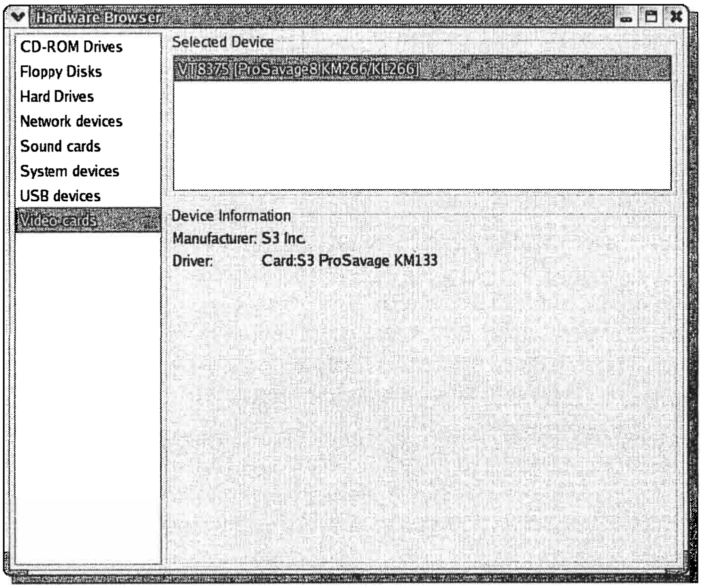


FIGURE 2-1: Use the Hardware Browser window to see the components you have.

Along with an indication of each type of device, you can see the Linux driver that is being used for the device. From this window you can find out the type of CD-ROM drive it has, the hard disk space, network card (Ethernet), sound card, and video card.

If you prefer to use the command line, you can run the `lpsci` command to get similar information.

Running the lspci command

The `lspci` command shows all PCI busses on your computer, as well as components that are connected to those buses. The bus is used to directly connect components (memory, video cards, network cards, and so forth) to the processor. Here's an example of output of the `lspci` command when I ran it on one of my computers:

```
# /sbin/lspci
00:00.0 Host bridge: Intel Corp. 82810E DC-133 GMCH \
     [Graphics Memory Controller Hub] (rev 03)
00:01.0 VGA compatible controller: Intel Corp. 82810E DC-133 CGC \
     [Chipset Graphics Controller] (rev 03)
00:1e.0 PCI bridge: Intel Corp. 82801AA PCI Bridge (rev 02)
00:1f.0 ISA bridge: Intel Corp. 82801AA ISA Bridge (LPC) (rev 02)
00:1f.1 IDE interface: Intel Corp. 82801AA IDE (rev 02)
00:1f.2 USB Controller: Intel Corp. 82801AA USB (rev 02)
00:1f.3 SMBus: Intel Corp. 82801AA SMBus (rev 02)
01:01.0 Ethernet controller: Intel Corp. 82557/8/9 \
     [Ethernet Pro 100] (rev 08)
01:07.0 Multimedia audio controller: Ensoniq ES1371
     [AudioPCI-97] (rev 06)
```

When I first booted the computer to setup mode, I learned the following about my computer (of course, your information will be different):

- **Motherboard** — Intel CA801E Desktop Board
- **Processor type and speed** — Intel Pentium III Processor (650 MHz)
- **RAM** — 128MB of System RAM

With the output from `lspci` or the Hardware Browser window (shown previously), I could also have learned about some of the other components:

- **Video card** — Intel 82801E DC-133 CGC graphics controller (built into the motherboard)
- **Network card** — Ethernet Pro 100
- **Sound card** — Ensoniq ES1371 sound card

The hardware shown here meets the minimum requirements for installing Red Hat Linux with a GUI on the computer. The 128MB of system RAM might not be enough to effectively run some of our video projects (192MB is recommended for a GUI install).

Note Hardware information used by `lspci` and `hwbrowser` is taken from the `/usr/share/hwdata/pci.ids` file. It is possible that more recent hardware may not be detected. You can get an updated version of this database from the Linux PCI ID Repository (`http://pciids.sourceforge.net/`).

Checking log files

The `/var/log` directory is another place to look for information about the hardware on your computer from Linux. In particular, you can look for failures to probe or load modules for the hardware on your computer. Check out the following files in that directory:

- boot.log — Shows if start-up and shutdown of different system services were successful.
- dmesg — Contains messages from the kernel ring buffer, to see messages that occurred during the boot process. You can see names and addresses of the hardware devices it finds and modules that are loaded.
- messages — Contains more detailed messages of processing that occurs during system startup. This file also continues to gather information about processing that goes on while the system is running.
- XFree86.0.log — Includes details about the processing of your XFree86 video driver.

Armed with the information I just described, I have most of what I need to evaluate whether or not my computer will work for Linux Toys. I can also use this information if I have problems and need to ask for help from a Linux mailing list or support organization.

Of course, if you don't have Linux installed yet, you may have to boot Windows to find out hardware information about your computer.

Evaluating hardware in Windows

If you currently have a Windows operating system running on your computer, you can view information about your computer hardware from a Device Manger (or similar) window. For example, here's how you can see hardware information in Windows XP:

1. From the start menu, you could select Control Panel.
2. Open the System icon. The System Properties window appears.
3. Click the Hardware tab.
4. Click the Device Manager button. The Device Manager window appears, as shown in Figure 2-2.
5. Click any of the items listed (display adapters, network adapters, processors, and so on) to find information about that item.

With information about your computer, you can query the Red Hat Linux Hardware Compatibility List (http://hardware.redhat.com/hcl) to see if your hardware is supported. You can also use a search engine (such as google.com) to check whether your hardware is supported (try searching for the name of your hardware followed by the word Linux).

If you feel like the computer you have might work fine with Linux, you can skip the rest of the hardware descriptions in this chapter and either continue to the section on finding open source software or go right to the project you want to start. The next several sections, however, will help you if you want to:

- Go out and buy a PC for the projects
- Upgrade selected pieces of your computer to make it run better for the projects

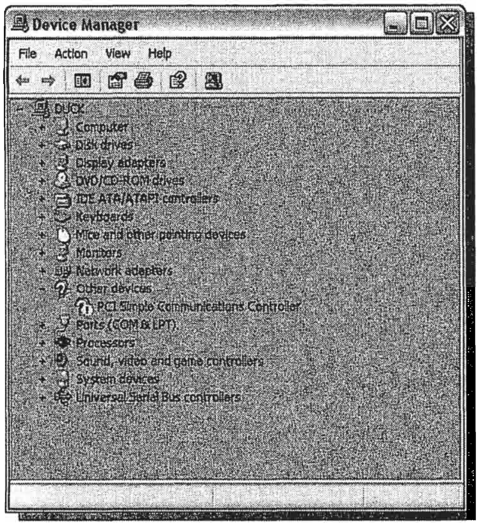


FIGURE 2-2: In Windows, use the Device Manager window to find your hardware.

Tip

Before getting a computer with the intention of upgrading it to work with Linux, I recommend you confirm that the computer can be upgraded. With some cheap computers, you're stuck with the video and sound that comes on the motherboard. There may be no AGP slot for a video card and no extra PCI slots to add features you need. RAM slots may be limited as well.

How Do I Choose . . .

The various Linux Toys projects can place different demands on your computer. Some demand lots of disk space (such as video and audio projects). A project that requires a graphical, rather than just a command-line, interface will work better with a better video card and more RAM. Going through the sections that follow will help you choose the components that will work best with your project.

. . . the case?

Functionally, most any PC computer case (the metal or plastic that houses the computer) will work for most Linux Toys. Aesthetically, you might want to be fussier about the look and feel of the box you use. This is especially true if you want to put the box on an entertainment center. Here are a few things to consider:

- **Form factor**—The form factor of the motherboard can affect the size of the case and the arrangement of connectors. If you are hunting around for a PC to go on your entertainment center, you might look for a desktop system that is meant to go horizontally. As you can see in Figure 2-3, not every case will look good with your other entertainment components.

FIGURE 2-3: While most PC cases will do for Linux Toys, some look just plain ugly.

Today, ATX is the most popular form factor for desktop PCs. Older AT form factor motherboards might have older ISA slots (making it more difficult to get modern sound, video, and other cards to use with it). The FlexATX style motherboard is a fairly new design, created for smaller cases that are easier to carry around or fit into home-entertainment areas.

- **Noise** — Fans can be noisy. Check out the noise level on a computer before you set it up as a component on your entertainment center. A way around the noise problem is to put the computer near your wiring closet and operate it remotely (for whole-house stereo or video). The noise level may be less critical when the PC isn't sitting in your living room.
- **Expansion** — If you are using the computer to build the "Be Your Own ISP" project, make sure the computer (or computers) you are using has enough slots to hold the cards you need. You might need multiple slots for several modems or TV cards. Also, as I mentioned before, PCI slots might be more useful than the older ISA slots.

Note

PCI slots support 32- and 64-bit cards. They are white or light tan. ISA slots are black, while EISA slots are brown. PCI slots are about 3.5" in length and ISA slots are about 5.5".

. . . a CPU?

The processor needs to be at least a Pentium 200Mhz processor for text mode and 400Mhz for GUI mode. Old 386 and 486 processors will not work with the latest Red Hat Linux. Pentium chips have been around since about 1993, so that still gives you about 10 years' worth of old computers to use.

Note

One exception is the "Linux on a floppy" project described in Chapter 13. That project will run a mini version of Linux on most computers that have a floppy disk (including older 486 computers).

Pentium III processors came out in 1999 at 450MHz and 500MHz speeds. If you can spring for the extra money, a Pentium III will give you better audio and video performance than earlier Pentiums. Pentium IV processors are available today at speeds of over 3 GHz. (No wonder there are so many computers around for us to make toys out of!)

CPU chips from Advanced Micro Devices (AMD) are also acceptable. When Intel dropped the 386, 486, and so on naming convention (which they couldn't copyright) and went over to the Pentium name, AMD followed by naming their processors K5 (for Pentiums), K6 (for Pentium II) and so on. Supported AMD CPUs include K5, K6, K6-2, K6-3, Athlon, Athlon XP, and Duron.

All of our video projects (Chapters 4, 5, and 8) require faster processors than are needed for other projects. We recommend at least a 450 MHz processor for those projects.

. . . a hard disk?

The hard-disk size you need for Linux Toys varies from project to project. Disk-space requirements for some projects are based on how much audio, video, or other content you add to it.

Hard-disk space requirements are listed with each project. Here are some general guidelines you should follow as you are figuring out how big a hard disk you need for Linux Toys.

- Projects that require a GUI will need 1.7GB (for a Personal Desktop) or 2.1GB (for a Workstation) Red Hat Linux installation. You can reduce this by removing unneeded packages, but you might run into dependency problems (i.e., another package might need the one you removed to make it work).
- A project that doesn't require a GUI can begin with a Red Hat Linux Server (850GB) or minimum (475MB) installation. In many cases, however, you need to add some packages that are needed by the projects.
- If you are someone who just has to have everything, an Everything installation of Red Hat Linux requires about 5.2GB of disk space.

On top of Red Hat Linux, you need room for Linux Toys software. Some projects require little or no additional software. Others (the music jukebox, for example) require a lot of space for adding software and possibly much more for adding your content (such as music CDs or video content). Check each project's requirements just to make sure.

You can use multiple hard disks, if you like. The trick is, however, that each disk must be on its own mount point. So you need to make sure that you don't run out of disk space where you need it. For example, having two 4GB hard disks doesn't ensure that you can do an Everything install if you installed one partition as the root (/) partition and the other as the `/var` partition.

Also, all hard drives are not created equal. Hard drives that do not support direct memory access (DMA) or are unable to operate in programmed input/output (PIO) mode 4 will not be able to effectively record or play video. Refer to Chapters 4 and 5 for some insights into hardware demands for video recording and playback.

Chuck's Tech Tip

Logical Volume Manager (LVM) can be utilized to make disk management far more flexible. LVM allows you to turn individual disks into what are called *physical volumes*. These physical volumes can then be combined to create *logical volumes*. Logical volumes are the same as disk partitions and can be formatted and mounted to the main file system as needed. You should exercise caution when using LVM, however. A volume made up of multiple disks can be irreparably damaged if one of the disks that make up the logical volume crashes.

. . . RAM?

Random Access Memory (RAM) is another area where the requirements are much higher if you are running the Red Hat Linux GUI. RAM requirements for Red Hat Linux are:

- **Mimimum (text mode)** — 64MB of RAM
- **Mimimum (GUI mode)** — 128MB of RAM
- **Recommended (GUI mode)** — 192MB of RAM or more

Will the computer run on less than the minimum amounts of RAM? It probably will, but you won't be happy. When the computer runs out of RAM, it tries to store some of its critical data in swap space (on your hard disk). If your computer starts swapping every time you open or move a window, your performance suffers drastically.

RAM has come down so much in price over the past few years that you should consider adding RAM (the more the merrier) to your computer. Here are a few suggestions about how you can go about finding the RAM chips to fit your computer:

- **The manual** — If you still have it, your computer's manual should describe the type of RAM needed by your computer.
- **The Web** — There are a bunch of Web sites that help you determine and (in many cases) purchase the RAM that is appropriate for your computer. Here are a few locations that have good tools for finding the RAM you need:

  - `www.4allmemory.com` — Enter the manufacturer and model number to find RAM that will work on your computer. Besides telling me what RAM was available and its cost, it told me the standard memory, maximum memory, and the number of expansion slots on my computer.
  - `www.kingston.com` — Besides letting you search for RAM by manufacturer, this site lets you search by memory type, model name, and part number.
  - `www.memorystock.com` — If you know the type of RAM you need, select it from the categories in the left column of this site. Otherwise, you can do keyword searches for your computer's model number.
- **Bins and old computers** — You might be able to salvage the RAM you need from a bin at a used computer store. Or you could cannibalize another old computer to pull out its RAM.

If you go the route of purchasing RAM on the Web, search around a bit. Especially with some of the older RAM chips, I've noticed that there can be a pretty wide range of prices for the same chip.

Before you purchase extra RAM, also make sure that your PC can handle it (at an economical price). I have an old PC that has four slots for memory, but each slot can take up to 32MB RAM chips only. Those 32MB chips cost nearly $40 each at a few Web sites I checked. So it would cost me $160 just to make the computer into a usable system.

. . . a video card?

Support for different video cards in Linux has improved dramatically over the past few years. However, you can still get a clunker that either won't work or will work poorly.

If you can't get your video card to work in Red Hat Linux, you should consider buying a cheap card that will work for you (as opposed to hunting for a workaround or a driver). You can find older supported video cards from any used computer store.

Drivers for video cards in Red Hat Linux are maintained by the Xfree86 project (`www.xfree86.org`). From that site, click the link to Support, Documentation and Resources. From there, you can search their database for information on your video card or click links to find supported hardware.

Chuck's Tech Tip

If you are looking for a video card that will work under Linux every time, I strongly recommend any video card that carries the NVIDIA chipset. NVIDIA allows you to download the source-code drivers directly from its Web site so you can compile them for whatever system variation you are working on. While I always give preference to open source developed software, drivers are among those rare things that are best written by engineers who have an intimate knowledge of the hardware itself. NVIDIA is the only video-card manufacturer I am aware of that provides Linux drivers written by its own engineers. (NVIDIA did not pay for this plug; I am simply a happy NVIDIA video chipset user.)

Once you have installed Red Hat Linux, you can configure your video card using the `redhat-config-xfree86` command (while you are logged in as the root user).

. . . a sound card?

As with video cards, it's often easier to just buy a working, used sound card if the one you have isn't working. I go into your options for getting a sound card in Chapter 3.

. . . other hardware?

I go through other specific hardware requirements for Linux Toys in each chapter where the hardware is used. Other hardware needed by various Linux Toys projects includes modems, video cameras, TV cards, relay boards, and Ethernet cards.

Using Kernel Hardware Information

If you install the kernel-source RPM package that comes with Red Hat Linux, you can see all the code for the drivers and modules that are in Linux. Part of what comes in that package is a bunch of documentation that describes what hardware is and is not supported.

When the kernel-source package is installed, you can find the Linux source code in the `/usr/src/linux-2.4` directory. In particular, you can find hardware documentation in the `/usr/src/linux-2.4/Documentation` directory. Table 2-1 lists a few interesting files and directories in that directory. To view the contents of those directories, change to the Documentation directory and page through each file. For example:

```
$ cd /usr/src/linux-2.4/Documentation
$ less kernel-docs.txt
```

Press the spacebar to step through each file. Type **q** to quit.

Note If the kernel-sources package is not installed, you can install it from the second CD in the Red Hat Linux installation set. See Appendix C for information on installing Red Hat Linux packages.

Table 2-1 Notable Directories and Files in /usr/src/linux_2.4/Documentation

File/Directory	*Explanation*
`cdrom`	This directory contains files that describe different CD-ROM drives that are supported. The ide-cd file contains some good information for debugging CD-ROM problems (including what to do if your CD-ROM isn't detected at boot time and what to do if you get timeout/IRQ errors).

File/Directory	*Explanation*
`Configure.help`	This file contains descriptions of each known driver in the Linux kernel that are used with tools that reconfigure the kernel. You can search this file for the particular piece of hardware you are interested in. This can help you find the driver name (which you can use if the module or driver isn't being loaded automatically) and links to more information about the hardware.
`devices.txt`	Devices (represented by files in the `/dev directory`) provide access points to the hardware on your computer. This file contains the official device numbers and names for each device available to be used by Linux.
`ide.txt`	Use this file for suggestions on how to debug you IDE devices (such as hard disks and CD-ROM devices). In particular, it includes IDE parameters you can pass to the kernel at boot time. This can help Linux find devices that it was unable to probe for itself.
`kernel-docs.txt`	Contains information about books and Web sites that describe how to understand and write components for the Linux kernel
`LVM_HOWTO`	If you want to create a large pool of disk space from several different disk partitions, this HOWTO tells you how to do that from the command line. (The easier way is to create LVM partitions during Red Hat Linux installation.)
`memory.txt`	Contains some tips for dealing with memory problems
`modules.txt`	Describes commands for loading and getting information about modules. If certain hardware is not detected, you can use the commands described in this file to manually load the modules you need to access the hardware.
`networking/`	Files in this directory describe networking cards that are supported in Linux. This is a good place to start if your Ethernet card isn't being detected.
`sound/`	Use the files in this directory to get information on supported sound cards. The Introduction file has really good information about adding and checking sound cards, as well as some troubleshooting tips.
`usb/`	Files in this directory describe supported USB devices, including scanners, Webcams, and Bluetooth devices (to name a few). You can check the Linux USB site (`www.linux-usb.org`) for more information.
`video4linux/`	This directory is important for finding Webcams and video cards that are supported in Linux. The Cards file in the `bttv` subdirectory lists all supported video capture cards. I strongly recommend checking this list before you purchase a Webcam or video capture card. (Better yet, use video capture cards and other hardware we recommend for best results.)

Just reading through the Documentation files can give you great insight into how Linux works. If the inner-workings of the kernel fascinate you (aside from what you need to build Linux Toys), reading these files is a great place to start.

Shopping for a Used PC

Buying a used computer can be like buying a used car. You can end up broken down on the edge of the information superhighway, without a tow truck in site. For a $20 investment, you might take a risk on a computer that (if it doesn't work) will end up in the dumpster. If you start getting into the hundreds (approaching the costs of a low-end new computer), you better have some recourse if it doesn't work.

So, with so many used computers out there, how should you approach this task? Whether you are looking at a garage sale, auction, computer store, or Web site, you want to balance the risk you are taking with the amount you are paying.

Questions to consider

You don't have to blindly assume all the risk yourself when you shop for a used computer. Here are a few basic questions you should ask about a used PC before you buy it:

- **Can I try before I buy?** If you can't try it and you can't return it, don't pay much for it. There are too many used computers around to risk more than a few dollars on a computer you'll have to throw away if it doesn't work.

  If you can try it, go through the evaluation process described earlier to at least know if it boots and what its basic components are. It's worth a few more dollars if you at least know that it starts up and contains the minimum you need to run Linux.

- **Can I return it if it doesn't work?** Return policies on used computers vary widely. If you can't see the PC before you buy it, at the very least, you would like a Dead On Arrival (DOA) agreement that lets you return it if it doesn't work once you get it home.

- **Can I have a warranty?** I've seen brand new computers for only a few hundred dollars that have three-year parts and labor warranties. There are rebuilt and recertified computers from reputable retailers available for under $300 that offer at least a three-month parts and labor warranty. If you are paying more than that for a used PC, you should have some kind of warranty if it doesn't work.

If you are going to take the plunge and buy a used computer that you know you can't return, you might at least check whether the components in the computer might be salvageable. A whiz-bang video card, decent 56K modem, or an extra Webcam thrown in can soften the blow if the computer has a bad hard disk or blown power supply.

Where to buy?

There are used computers everywhere. If the one in your closet won't work, here are a few places to look.

Computer stores

Stores that sell new computers also sometimes sell used computers. A big advantage to these stores is that, like a car dealer, they may want to move out their used inventory before it further

loses its value. They also will have the expertise to check that a computer is working, and will meet your needs, before you buy it.

Auctions

We went to an auction where they were liquidating the contents of a computer supply store. Here are a few tips for buying at auction:

- **Know the value** — It's easy to get caught up in the excitement of an auction. Before you start bidding, know what a computer is worth and set a limit on what you will pay.
- **Bring your LUG** — There were about a dozen members of our Linux Users Group at the auction Chuck and I attended. By standing between a couple of guys who buy a lot of computers, I could get a good sense of what comparable new computers would cost.
- **Look in the pile** — Instead of trying to sell a bunch of old computers separately, there may be a pile of computers that the auctioneer sells as one lot. If you like to tinker, there might be a jewel in this bunch or some parts you can use to fill what you are missing. The group of computers shown in Figure 2-4 went for $15 at the auction.

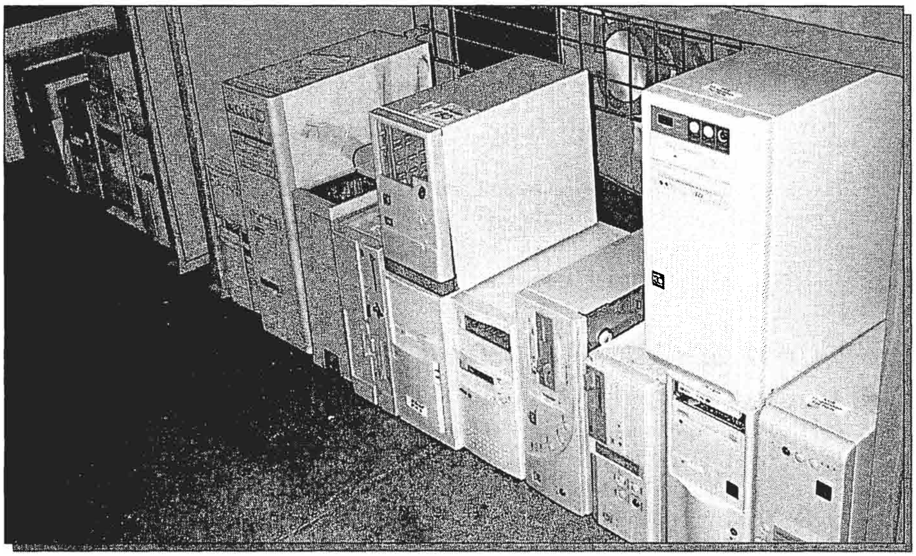


FIGURE 2-4: Is there a working computer in this bunch?

If you don't like to put together computers or fix them yourself, there are often new computers for sale at these auctions as well. If you pay anything near retail prices, however, find out if a manufacturer's warranty applies and if you can get a DOA guarantee from the auctioneer (this auctioneer gave bidders a day to get it home and check if it worked).

Surplus stores

Large companies often have to upgrade their computers to get the software they need to work. In our area (western Washington state), The Boeing Corporation has a surplus store where you can find good deals on used computers.

PC manufacturers (returns and rebuilts)

While not the cheapest way of buying a computer, buying returned and rebuilt computers is a good way to get late-model computers for less than late-model prices. For example, Tiger Direct (`www.tigerdirect.com`) sells "factory recertified" computers for less than comparable new ones would cost. Some have manufacturers' warranties, but Tiger Direct will give you a 30-day limited warranty if the manufacturer's warranty is expired.

Finding Open Source Software

To build the projects described in this book, we expect you to get Red Hat Linux. We'll give you all the other software you need (on the Linux Toys CD).

Appendix C tells you ways of getting Red Hat Linux for free or for cheap. The Linux Toys CD comes with this book. So the larger issues when it comes to getting other software are:

- Where do I find out about enhancements to Linux Toys software?
- Where do I find software and information about other open source projects?

If you are happy building the basic Linux Toys described in this book (and you don't happen to run into any problems along the way), you don't need to read this section. But if you want to go above and beyond the confines of this book, read on about where to find open source software and the people who make and use it.

Where open source people gather

There are open source people all over the world (unfortunately you can't always spot them in public). They spend a lot of time in front of their computers. That's why by far the best way to connect to the open source world is over the Internet. If you'd prefer to meet people in person, however, I'd suggest tracking down your local Linux Users Group. Here are some suggestions on finding Linux people that you can get help from, give help to, or just pal around with:

Mailing lists

Every major topic associated with Linux has at least one mailing list available where you can get or offer help on that topic. Most of these lists let you either have messages sent right to your e-mail address or access an archive of past messages that you can go through.

Red Hat Inc. maintains a list of popular mailing lists you can join at `www.redhat.com/mailman/listinfo`. Usually, the best way to join a mailing list, however, is to go to the

project site that interests you. Here are some links to mailing lists that you can use to get different Linux features and hardware discussions:

- **Sound Cards**—`www.alsa-project.org/mailing-lists.php3`
- **Video Cards**—`www.xfree86.org/mailman/listinfo`
- **Webcams**—`www.redhat.com/mailman/listinfo/video4linux-list`
- **Web Server**—`www.apache.org/foundation/mailinglists.html`
- **CDDB**—`www.freedb.org` (click on Mailinglists)

Linux Users Groups

In the past few years, Linux Users Groups (LUGs) have been popping up all over the world. Chances are, there is already one not far from where you are. LUGs provide a great way to learn and just have fun with people who are interested in Linux.

For example, Chuck and I belong to the Tacoma Linux Users Group (`www.taclug.org`). The group offers a monthly meeting that includes lectures, demonstrations, and workgroups. We all keep in contact using the TACLUG mailing list and IRC chat. Kevin (our technical editor) is president of the Kitsap Peninsula Linux User Group (`www.kplug.org`).

Red Hat, Inc., maintains a list of LUGs and information on how to create your own LUG at `www.redhat.com/apps/community`. As I mention in Chapter 1, Linux Online also has a list of user groups (`www.linux.org/groups/index.html`). Or just search for "Linux User Group" and the name of your town or city at `www.google.com` and see what you get.

Where open source projects live

You can find many open source projects at FreshMeat (`www.freshmeat.net`) and SourceForge (`www.sourceforge.net`). Both sites allow you to do keyword searches to find projects that interest you. For open source projects related specifically to Linux Toys, you can go to the Linux Toys Web site (`www.LinuxToys.net`).

Summary

Linux Toys projects will run on many PCs built in the past 10 years. Although you can install Red Hat Linux and the Linux Toys described in this book on the latest hot PC, I've tried to give you some advice in this chapter on how to search out a cheaper computer that will work well for many of our projects.

After you have hunted down your hardware, this chapter describes where you can find additional software to add to your Toys. It also describes where to find mailing lists and Linux User Groups (LUGs) to get more information on projects that interest you.

Entertainment Projects

part

II

in this part

Making a Music Jukebox

chapter 3

in this chapter

☑ Setting up the music jukebox

☑ Ripping and playing music

☑ Using a CD database

☑ Playing songs remotely

☑ Creating playlists

☑ Checking for problems

Even back in the days when playing music on my computer meant popping in a CD and listening on some dinky little speakers, I thought it was fun to do. Now, with the audio tools in Linux (and a few others we added) you can do a whole lot more, such as:

- Copy your music CDs to hard disk (called *ripping*)
- Store the music efficiently, using open source compression techniques
- Gather title, artist, and other information about each CD and song
- Play back your songs individually, from a playlist or at random
- Hook up your computer to your stereo equipment to get higher-quality sound

This chapter describes how to build a music jukebox in Linux on your PC. The basic project is a stand-alone CD ripper/player that continuously pumps out the music you add to it. With the enhancements described later, you can use the same jukebox to be a public Compact Disc Database (CDDB) server and play songs in different ways.

About the Linux Toys Jukebox

The Linux Toys Jukebox (ltjukebox) project is a simple CD ripper and player, built on some cool technology. At its core is the Red Hat Linux operating system, with some music compression/playing tools (we chose Vorbis compression and the Ogg file format) and the CDDB database (in our case, CDDB from `freedb.org`) built on top.

The basic PC for this project requires a sound card, a CD drive, and a large hard disk (I'll expand on this a bit later). A keyboard and monitor are optional once the jukebox is set up. However, you might want to add a network card (or keep your keyboard/monitor connected) to make changes to your collection or CDDB database going forward.

Once you have the basic Linux Toys Jukebox up and running, you can expect it to do the following:

- Rip all songs from any CD you insert into the player (provided the CD is among those listed in the CDDB) without any interaction
- Compress each song
- Create a directory (named for the CD) and store each song in that directory
- Automatically play, in random order, all songs from all CDs you ripped

We designed the player to be up and running quickly and to play automatically. Think "commercial free radio." It simply plays and plays with no interaction required by the end user. But think about this: After you've set up the jukebox, you'll have a database full of CD information and a hard disk full of identifiable music. From that point, there are an unlimited number of ways you can enhance it.

Figure 3-1 shows what a completed Linux Toys Jukebox can look like after you have hooked it into your stereo system. No flashing lights, but the basics are in there.

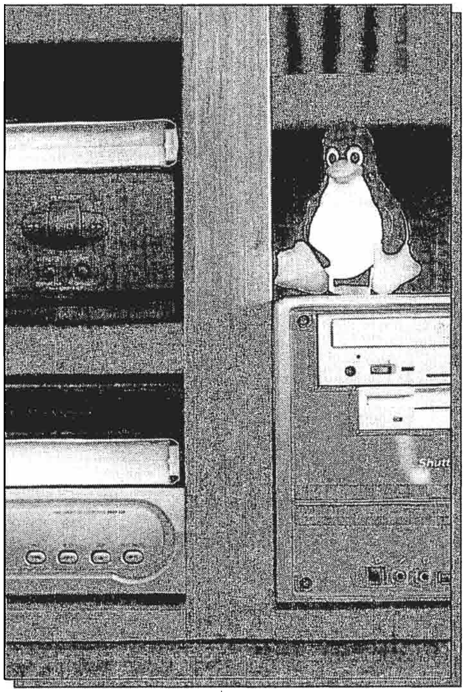


FIGURE 3-1: Store and play your music with a Linux Toys Jukebox. (Shuttle XPC image used with the permission of Shuttle Inc.)

Of the software in this project, the freedb.org CDDB is by far the largest component added to Red Hat Linux to make the jukebox. This database includes hundreds of thousands of entries identifying music CDs, including artists, albums, and song titles. In fact, much of the disk space needed to make this a stand-alone player is used for this database (more than 2.5GB).

If you want to save some disk space, and your PC has a connection to the Internet, you can skip installing the CDDB and just have your jukebox query a remote CDDB server. Use of the CDDB service and the database itself are covered under the GPL, so you can use it without cost.

Note

Ripping doesn't mean ripping off. You should never use the Linux Toys Jukebox, or any other CD ripper for that matter, to copy CDs that you don't own or to distribute copies of CDs to others. The jukebox should be used only to privately store and play music that you own.

Configuring the Linux Toys Jukebox

The general steps for configuring and using your Linux Toys Music Jukebox are as follows:

1. Gather hardware.
2. Install and set up Red Hat Linux.
3. Install the Linux Toys software.
4. Start the Jukebox.
5. Load the music.
6. Play, play, play!

If you are ready, you can start by gathering the stuff you need to make the Linux Toys Jukebox.

Step 1: Gather your hardware

Although most modern PCs will work fine for this project, a large hard disk and sound card are required for the basic Linux Toys Jukebox. Another consideration is whether or not you want a finished case for the player to fit nicely on an entertainment center. Here's a rundown of what you need:

The personal computer

Most PCs that are compatible with Red Hat Linux should work for this project. You should check minimums relating to CPU and RAM in Chapter 2 and Appendix C. As I mentioned earlier, you might want a form factor that fits on a stereo shelf. Figure 3-2 shows an example of a small form factor computer that will fit in an entertainment center.

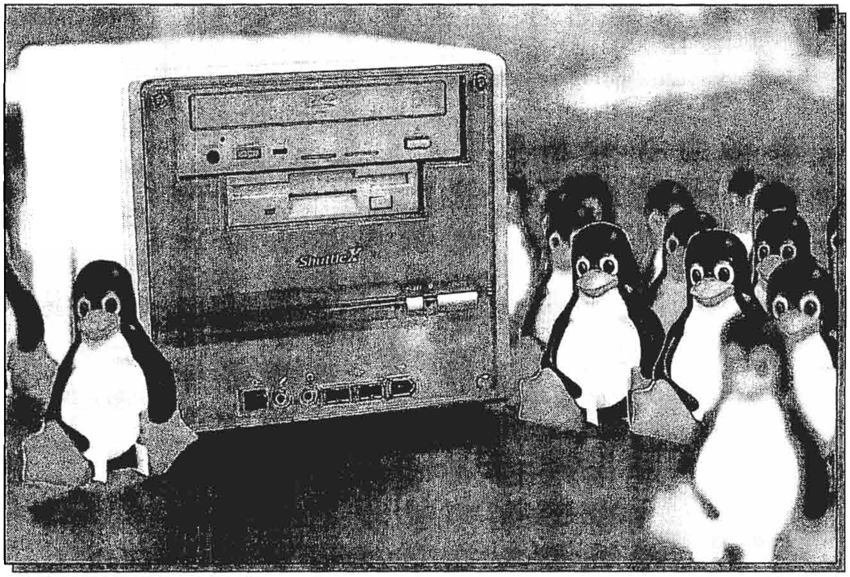


FIGURE 3-2: A small form factor helps the jukebox blend in better. (Shuttle XPC image used with the permission of Shuttle Inc.)

The computer shown in Figure 3-2 is the Shuttle XPC system. You can get one for under $1000. Its look makes it more attractive to put in an entertainment center or anywhere in your living areas. Our technical editor on this book, Kevin, bought a used set-top box that is just a PC on the inside. The box looks like a VCR, but runs with Linux on the inside.

There are some hardware requirements for the PC you use with this project.

Hard disk

To make this project self-contained, you will put the whole freedb.org CDDB on the computer. You also have to leave a bunch of space for Red Hat Linux and, of course, the music itself. Here is the basic breakdown of the disk space you will need:

- **Red Hat Linux** — We recommend a Personal Desktop install, which requires about 1.5GB of disk space. (You can get by with a Minimal install, plus a few packages we recommend for under 500MB. The default jukebox will work fine, but you will have no GUI and you will have to work out some package-dependency issues.)
- **ltJukebox RPM and freedb** — The amount of disk space needed by the ltJukebox package is less than 1MB. Two songs we provide to test the jukebox take up about 6.2MB. However, the thousands of entries in the freedb database require about 3GB to install. If you have a network connection on your jukebox, you can skip installing the freedb database.

- **Music**—The size of each music CD varies. In my unscientific test of about a dozen popular CDs, I averaged about 60MB per CD (varying between 40MB and 100MB) of compressed music. At that rate, I could store about 17 CDs per GB.

So here are our recommendations for the amount of disk space you should have available for this project:

- **Basic Jukebox**— At least 6GB. With this, you can install the whole CDDB, Red Hat Linux Personal Desktop install, and about 25 CDs. Each extra GB will get you space for about another 17 CDs.
- **Minimum Jukebox**—At least 2GB. You can do a minimal Red Hat Linux installation (with a few packages added) and not install the CDDB. You will need a network connection to get CDDB information from another computer, and you will only have 1GB of space (about 17 CDs) for music.

CD drive

You need the CD drive for loading your music CDs. CDs are also the preferred media for installing Red Hat Linux and the Linux Toys software.

Sound card

You need a sound card that is supported under Red Hat Linux. If your PC comes with a sound card, go ahead and try it. The firstboot program (described in Appendix C) lets you configure your sound card. Step 2 includes instructions on how to use `sndconfig` to detect and configure your sound card if firstboot can't do it.

Network card (optional)

Because the basic jukebox is designed to be self-contained, you don't need a network card. However, if you have a LAN at home or in your office, a network card gives you the option of using extended features, such as getting automatic CDDB updates or using the CDDB from another computer. It also saves you about 3GB of disk space.

The stereo (optional)

If your stereo can accept audio input, you can plug that stereo directly into your PC's sound card. I use the cable shown in Figure 3-3 to connect my sound card to my stereo.

Chuck's Tech Tip

It's important to note that this project was designed to be an appliance, requiring little or no interaction with the end user. While you are selecting your hardware, keep this fact in mind. If you are handy with a Dremel tool and metal snips, you might consider removing the guts from an old stereo and mounting all of your computer parts inside. The result would fit nicely into most stereo rack systems.

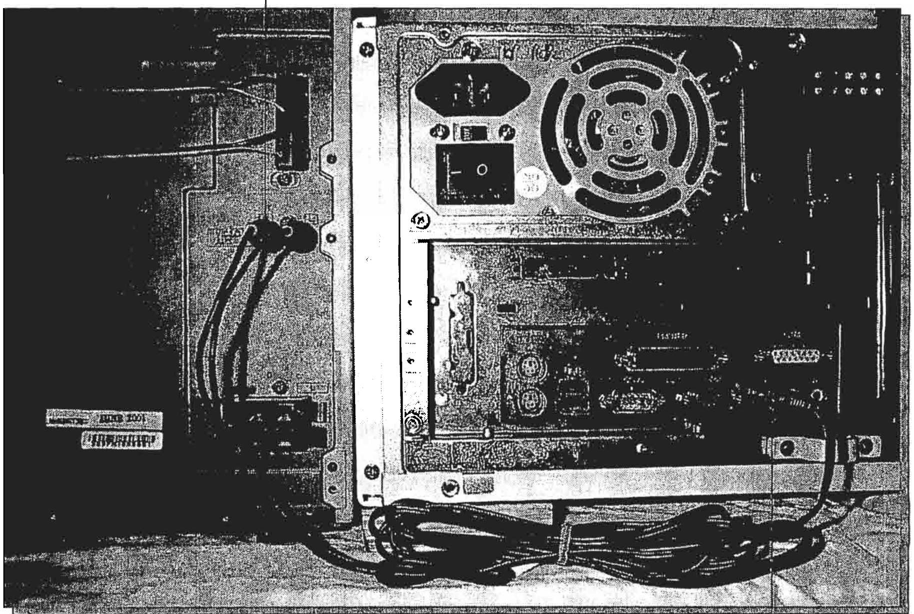


FIGURE 3-3: Plug your stereo directly into your PC's sound card.

Step 2: Install and set up Red Hat Linux

You need the three Red Hat Linux installation CDs to start. We recommend a Personal Desktop installation for this project (see Appendix C for information on getting the Red Hat Linux CDs and doing a workstation install). This will result in all necessary packages being installed for the project (except for those you add later from the Linux Toys CD).

The other option is to do a minimum install (select Custom during installation; then select Minimum). With a minimum install, you need to choose to select individual packages and add the packages shown in the next section.

Install additional packages

If you decide to go with a minimum installation, you need to make sure that at least the following packages are added to the minimum install:

- **vorbis-tools** — To compress and playback the music files
- **cdparanoia** — To rip the CDs to hard disk

- **kscd** — A CDDB-enabled CD player to add new CDs to your database. (You need to have a GNOME or KDE GUI installed to use this application.)
- **eject** — To eject the CDs after they are ripped
- **aumix** — To control the volume
- **sndconfig** — To detect the sound card. (This is a nongraphical equivalent to redhat-config-soundcard.)

Cross-Reference

See Appendix C for information on using the Add/Remove Applications window or `rpm` command to install packages if you need to add these packages after initial Red Hat Linux installation. You may also have to install some packages that these packages depend on.

Configure your sound card

Once Red Hat Linux is installed, if the firstboot program is unable to configure your sound card, you can try to configure your sound card using the `sndconfig` program:

Note

If you are new to Linux, at this point you need to understand how to get to a shell command line (the usual way from a graphical screen is through a Terminal window) and what a root user is. If you don't know those things yet, refer to Appendix B for a primer.

1. Plug speakers, headphones, or a stereo amplifier into the Audio Output port of your sound card.
2. Login to Red Hat Linux (either login as root or have the root password available).
3. Use either of the two tools for configuring your sound card (if one doesn't detect the card, try the other):
   - From the desktop menu (the red hat), click System Settings ➔ Soundcard Detection. The Sound Card Configuration window appears and tries to detect your sound card. Follow the instructions to configure and test your card. If no sound card is detected, try the next tool.
   - From a Terminal window, type `/sbin/sndconfig`. Follow the instructions to detect and try your sound card.
4. If the card seems to be detected, but you don't hear any sound, try the following:
   - Check that the speaker is plugged into the right port on your sound card.
   - Check that the volume is turned up on your speaker (to a reasonable level).
   - Run the following command, to make sure that maximum volume is being sent to the sound card: `aumix -v85`. (You can go as high as 100, but it might sound distorted on a cheap sound card.)

5. If neither configuration tool can properly configure the card, you can do one of the following:
   - **Install a new card** — There are lots of cheap, compatible sound cards around. If you are new to Linux, it may save you time hunting for and compiling the right driver. You can get a brand new sound card for about $30 that will work great; used ones can cost just a few dollars. If you want to go for a used card, check the Supported Hardware list in the Linux Sound HOWTO (`www.linux.org/docs/ldp/howto/Sound-HOWTO`). The list is a bit outdated, but it may be just what you need to pick up that bargain sound card from the local used-computer store.
   - **Check out ALSA** — The Advanced Linux Sound Architecture Web site (`www.alsa-project.org`) has the latest sound drivers available to use with Linux. Try the New Users link for a list of cards (by manufacturer) you can search for your card. If you find your card, follow the instructions for downloading, compiling and installing the driver.

At this point, you should have Red Hat Linux installed on your PC with a working sound card.

Step 3: Install the Linux Toys software

The Linux Toys Jukebox software you need to add to your Red Hat Linux system is available from the Linux Toys CD that comes with this book. The software is packaged in Red Hat Package Management (RPM) format, so it can be easily installed from Red Hat Linux. To install the ltJukebox RPM, along with two songs and a package that defines the CD information format, you can use the install.me script for ltJukebox.

To install the ltjukebox software, do the following:

1. Insert the Linux Toys CD into the CD drive.
2. From a Terminal window, as root user, type the following:

```
# mount /mnt/cdrom
```

3. Install the software as follows:

```
# cd /mnt/cdrom/ltJukebox
# ./install.me
```

Step 4: Add the CD database

If you have an Ethernet card installed on your computer with a connection to the Internet, you can refer to the "Using a Public CDDB Database" section later in this chapter to avoid the pain of installing your own CDDB. To install the `freedb.org` CDDB database so that your jukebox can be its own CDDB server, do the following procedure.

Caution Besides taking up a lot of disk space, the CDDB database takes a very long time to unpack and copy to your hard disk. A slow CPU and hard disk can make installing the CDDB a day-long process. If you don't mind taking the time, installing CDDB lets your jukebox be a standalone unit. You can add CDs manually that aren't in the database, as described later in this chapter.

1. Insert the Linux Toys CD into the CD drive.
2. From a Terminal window, as root user, type the following:

```
# mount /mnt/cdrom
```

3. Change to the `ltJukebox` directory, copy the compressed freedb database to `/usr/local/share/cddbdb` and uncompress it as follows (the copy and uncompress will each take a few minutes):

```
# cd /mnt/cdrom/ltJukebox
# cp freedb-* /usr/local/share/cddbdb/
# cd /usr/local/share/cddbdb
# bunzip2 freedb-complete*bz2
```

4. Untar the database by typing the following command.

Caution The next command you type will take up 3GB of disk space and run for several hours. If you don't have the space and time, you need to get them or reconsider your approach. Remember, the alternative is to have a connection to the Internet.

```
# tar xf freedb-complete*tar
```

While the database untars (is split off to thousands of little files from the one big file) you can go read *War and Peace* (or *Red Hat Linux Bible*), take a short vacation to the Bahamas, or finally learn to play the cello. If you are worried that it is taking a long time and are wondering if you have run out of disk space or hung, you can check every so often to make sure that the size of the `cddbdb` directory is growing by opening another Terminal window and typing the following:

```
# du -h /usr/local/share/cddbdb
```

You should see the size increasing of the directories being created. When the untar is done, you are ready to fire up your jukebox.

Step 5: Start the Linux Toys Jukebox

If you installed the Linux Toys Jukebox as described in the previous steps, you are ready to try it out. As I said earlier, the Linux Toys Jukebox is made to take over your CD drive and sound card (making it rather antisocial if you are not using it as a dedicated music player).

Caution

If you are using the GNOME desktop, you must turn off Magicdev before you proceed. Magicdev monitors your CD drive and tries to guess what to do with the CDs you put into it. Of course, that's trouble for this project, because you want the jukebox to take over the CD drive.

To turn off Magicdev from GNOME, do the following: Click the Red Hat menu button → Preferences → CD Properties. On the CD and DVD Preferences window that appears, turn off all check boxes that mount or play CDs; then click Close.

Any user logged in and using the GUI has to turn off Magicdev for the jukebox to work properly. If no GUI is running on the computer, there is no problem.

Once you start the jukebox, here is what it does:

- Starts playing, in random order, all songs already ripped to your hard disk
- Automatically rips and compresses the songs to hard disk from any music CD you insert into the CD drive. (It ejects any CD not listed in the CDDB.)
- If you installed the CDDB, starts the CDDB daemon to provide the song, album, and artist information the jukebox needs to store the music by album and song names. (If you didn't install the CDDB, the daemon will not run and the jukebox will check several different Internet sites to find the CDDB information it needs to rip your CDs.)

We installed two songs in the `musicd` directory to start off. So even before you start ripping CDs, music will begin playing and will play continuously until you stop the jukebox. The two songs are On the Porch and Listen (On_the_Porch.ogg and Listen.ogg). The songs were kindly donated by Chuck's friend Bill Broomall. To remove the songs from your music mix after the jukebox is running, as root user type:

```
# rm /usr/local/share/music/*.ogg
```

There are two ways you can start the jukebox:

- **Manual start-up** — This is good if you want the jukebox to run on occasion, but not all the time. To start the Linux Toys Jukebox immediately, type the following as root user from a shell (Terminal window):

  ```
  # /sbin/service ltjukebox start
  Starting musicd         [   OK   ]
  Starting ltautorip      [   OK   ]
  Starting cddbd          [   OK   ]
  ```

  When you are ready to stop the jukebox, type the following:

  ```
  # /sbin/service ltjukebox stop
  Stopping musicd         [   OK   ]
  Stopping ltautorip      [   OK   ]
  Stopping cddbd          [   OK   ]
  ```

Note

With no local CDDB database, the cddbd entries won't appear.

- **Permanent start-up** — If you are running the Linux Toys Jukebox as a dedicated music ripper/player (as it's intended), you can have the jukebox start up automatically every time you turn on the PC. The following command lists whether the jukebox is set to be on or off at different run levels:

  ```
  # /sbin/chkconfig --list ltjukebox
  ltjukebox  0:off  1:off  2:off  3:off  4:off  5:off  6:off
  ```

  Here, the jukebox is set to be off at all run levels. Next, type the following commands to set the jukebox to start automatically when you start Linux:

  ```
  # /sbin/chkconfig ltjukebox on
  # /sbin/chkconfig --list ltjukebox
  ltjukebox   0:off   1:off   2:on   3:on   4:on   5:on   6:off
  ```

  The first command turns on the ltjukebox permanently and the second shows that the service is on in all multiuser Linux run levels (2, 3, 4, and 5). The next time you restart the PC, the jukebox software starts running (or you can start it manually, as described previously, if you're not ready to reboot).

  To turn off the jukebox permanently, use `chkconfig` with the `off` option:

  ```
  # /sbin/chkconfig ltjukebox off
  # /sbin/chkconfig --list ltjukebox
  ltjukebox   0:off  1:off  2:off  3:off  4:off  5:off  6:off
  ```

  Now the ltjukebox service is off for all run levels (so it will no longer start automatically).

Step 6: Load the music

Once the jukebox is running, just insert a music CD into the CD drive. The drive will do one of two things:

- Start blinking happily as it copies and compresses music to your hard disk
- Immediately spit out the CD

If the jukebox is blinking, don't eject the CD. It may take a while to finish, so don't be in a hurry. When it is done ripping, the CD ejects automatically.

Note

If you eject a CD by mistake while it is ripping, don't worry. Just insert it again and it will start ripping from scratch. Over time, some files may accumulate in the `/tmp` directory. I'll describe how to get rid of those in the "Troubleshooting" section of this chapter.

If you have put a valid music CD into the drive and it immediately ejects, there is probably no entry for it in your CDDB. To add an entry into your local CDDB, as well as to the public Freedb.org CDDB, refer to the section "Adding CD information to the CDDB" for further information.

While the jukebox is running, it will continue to play all the songs from all the CDs you ripped. When it has gone through all the songs, it randomly plays all the songs again.

Playing with Your Jukebox

As it now stands, you could use the jukebox to:

- Continuously pipe music to your place of business
- Have hours of uninterrupted music for a party
- Listen to all your favorite music as you sit at your computer, building your next Linux Toy

With your music collection, a massive CD database, and a Linux operating system now on your computer, there are a lot of ways to take advantage of your jukebox. Here are some ideas:

Starting and stopping the jukebox

You can use the ltjukebox start-up script (located in the `/etc/init.d` directory) to start and stop the jukebox. As you can see from the examples in the step that starts the jukebox, there are three basic services started when ltjukebox launches: `musicd` (to play the music), `ltautorip` (to rip the CDs), and `cddbd` (to provide the CD database).

Step 5 of configuring your Linux Toys Jukebox describes how to start and stop the ltjukebox script (both manually and permanently). Here are a few options you can use while the jukebox is running.

The `status` option lists whether or not the jukebox is running and the process ID associated with each part of the jukebox.

```
# service ltjukebox status
musicd (pid
4201) is running...
ltautorip (pid
4211) is running...
cddbd (pid 4220) is running...
```

The `restart` (or `reload`) option stops and then restarts the jukebox.

```
# service ltjukebox restart
```

The `condrestart` option restarts the jukebox only if lock files exist for the service.

```
# service ltjukebox condrestart
```

A Lesson in . . . Linux Start-up Scripts

Most services that are meant to run continuously in Linux are launched from what are called *start-up scripts* or *run-level scripts*. To understand how these work, you need a few definitions:

- Run Level—Because Linux was made for big, multiuser computers, there was a need to have Linux be able to run at different levels of activity. By changing to different run levels, a Linux administrator can run the computer in single-user mode (1), one of several text-based multiuser modes (2 through 4), or multiuser with graphical interface.mode (5). There is also a run level that stops Linux (0) and one that reboots Linux (6).
- Daemon—A daemon is a process that runs continuously, in the background, waiting for something to happen. That "something" is typically a request that comes from a user or a program to use a service on your computer, such as: "Can I print a document, open a Web page, or send a mail message?" The daemon can respond to each request in many ways. It can accept the request, reject it, hand it to another program, or simply log that the request happened.
- Start-up script—A start-up script is a human-readable program intended to start a service (typically when your computer starts) and stop it (typically when your computer shuts down).

In Red Hat Linux, start-up scripts are stored in the `/etc/init.d` directory. You can type the `service` command with the `on` option to set the service to start when your system boots up. That default run level is typically 3 (if you are running in text mode) or 5 (in GUI mode). (The `service` command turns on the service for levels 2 through 5, to make sure it starts in all operational run levels.)

In the case of the Linux Toys Jukebox start-up script (ltjukebox), if you set it to be on, it runs commands that continuously monitor your CD drive (so it can rip any CDs you insert), listen for requests to the CD database, and play songs.

Although starting and stopping the jukebox while it is ripping can leave some processes and files behind (see the Troubleshooting section), for the most part it doesn't cause a problem. You can just start re-ripping the CD you started when the jukebox stopped and the CD will be ripped, compressed, and copied to the right place.

If you are interested in how start-up scripts work in Linux, check out the sidebar "A Lesson in . . . Linux Start-up Scripts." Otherwise, go ahead to the descriptions of the ltjukebox components.

Making your own playlists

Instead of just playing your entire collection randomly, you can create a playlist to select which songs to play. The playlist can be a simple text file that lists one of the three types of items, each on its own line:

- Song file — The full path name to a song. For example, `/usr/local/share/music/Joe_Smith-My_CD/01-My_Song.ogg`.
- Directory — A directory name causes all songs in that directory (and its subdirectories) to be played. For example, `/usr/local/share/music/Joe_Smith-My_CD` would play all songs from that CD; `/usr/local/share/music` would play all the songs from all the CDs placed in the `music` directory.
- URL — Plays a song that you identify on the Web. For example, a Web address for a song might look like: `http://www.linuxtoys.net/music/test.ogg`.

To create a playlist from your music collection, follow these steps:

1. Copy a list of your music to a file. For example, you could use the `find` command as follows:

```
# find /usr/local/share/music -print > /tmp/playlist
```

2. Use your favorite text editor to remove the songs you don't want, change the order of songs, or type new songs. To edit the file with the `vi` editor, you could type:

```
# vi /tmp/playlist
```

3. Copy the `playlist` file to `/usr/local/share/music` as follows:

```
# cp /tmp/playlist /usr/local/share/music/
```

4. Restart the jukebox by typing:

```
# /etc/init.d/ltjukebox restart
```

The jukebox will play the songs in your playlist in the order in which you enter them. To use a different playlist, copy it (it must still be called `playlist`) over the old playlist file. To go back to random playing of your whole collection, remove the playlist as follows:

```
# rm /usr/local/share/music/playlist
```

Once you create a playlist, you might want to save it in your home directory so you can use it again.

Chuck's Tech Tip

A useful addition to this project would be to have a dynamic playlist tied to a calendar. Holiday-specific songs would be randomly played within the normal stream of songs. As the holiday got closer and closer, the holiday-specific songs would be played with greater frequency. After the holiday passed, the songs from that holiday would cease to be played until the next year.

Playing songs manually

If you just want to listen to certain songs or all the songs from a particular CD, you can turn off the jukebox start-up script (`service ltjukebox stop`) and just run the `ogg123` player manually. The easiest way is to just use a file as an argument:

```
# ogg123 My_Song.ogg
Audio Device:   OSS audio driver output

Playing: 01-My_Song.ogg
Title: My Song
Artist: Joe Smith
Album: My Song Collection
Time: 00:00.01 [03:30.63] of 03:30.64(169.6 kpbs)Output Buffer
96.9%
```

Information about the file, the song it is playing, the artist, and album are displayed. You can also see the run times for the song as it progresses, its output rate, and how full the output buffer is. To play a whole directory of songs, use a directory name:

```
# ogg123 /usr/local/share/music/Joe_Smith-My_Song_Collection/
```

The previous example plays all songs, in order, from the directory shown. You can skip past the current song by pressing Ctrl+C as it plays. Press Ctrl+C twice quickly to exit from `ogg123`.

To play all the songs from a playlist, you could type the following (replacing `/tmp/playlist` with the name of your own playlist file):

```
# ogg123 -@ /tmp/playlist
```

Playing from other computers in your home

With your CD collection installed on your computer, you may also want to play your music from other computers in the house. As it is, you need to have a player on the other computer that:

- Can play OGG files
- Can play from a Web address

Just to give you an idea of how this might work, let's start with two Linux computers (the jukebox and another Linux system). Here's how you can set up the jukebox so that the other Linux system can play songs from it:

1. Add a LAN connection between your jukebox and the computer you want to share your music (as described in Chapter 7).
2. Set up the jukebox as a Web server (as described in Chapter 12).
3. Make your music catalog available to your Web server by linking it to a directory that you share from your Web server. For example, you could type:

   ```
   # ln /usr/local/share/music/ /var/www/html/music/
   ```

4. You could then use ogg123 to play a particular song from the directory via your Web server. For example, if your Web server were named abc.linuxtoys.net, to play My_Song from the Joe Hat album called Mine, you could type the following from another Linux computer on the network.

```
# ogg123 http://abc.linuxtoys.net/music/Joe_Hat-Mine/01-My_Song.ogg
```

A better, but less secure, way of sharing your music CDs with other computers on your LAN is to share the /usr/local/share/music directory using one of the following file-sharing applications:

- **Samba**—To share files and printers with Windows computers (and Linux computers as well), you can use the Samba file/printer sharing software that comes with Linux.
- **NFS**—To share files with other Linux computers, NFS is an excellent tool for connecting remote directories into a local file system over the network. You can actually mount /usr/local/share/music on the same directory of another Linux computer on your network.

Once you have shared your music directory on the LAN, another computer can play the song files as if they existed on the local computer.

Modifying Your Jukebox

Remember, this is your jukebox. You can do a lot to make it behave the way you want it to. If you don't like the way we configured the jukebox, you can reconfigure it. It you don't like the way an application works, you can modify and recompile the source code. Here are some ideas about ways you can modify your jukebox.

To be (a CDDB) or not to be?

The jukebox was made to run in standalone mode (that is, with no network connections). To make that happen, add the entire Freedb CDDB database of album, artist, and song information to the jukebox, and make it available to your local computer when you install the jukebox software. You may want to bypass this arrangement by using a public CDDB database or sharing your CDDB database.

Using a public CDDB database

Instead of installing the massive (nearly 3GB) CDDB database, you can choose to access a public CDDB database to download album, artist, and song information. If your jukebox has access to the Internet, this is a good way to go.

The Linux Toys jukebox will automatically detect that you do not have a CDDB database installed. Then it will automatically search several public CDDB servers to find the information it needs about your CDs. In other words, if you already have an Internet connection, you don't have to do anything to use a public CDDB database.

Sharing your CDDB database

You can do more with the CDDB database you have installed on your jukebox than use it to name the CDs you rip. You can offer it to others on your LAN to rip their own CDs and have information about their CDs pop up as they play.

By default, when the jukebox is on, the CDDB daemon process is running and able to share CD information with any computer that asks for it, provided a few criteria are met:

- Your jukebox is connected to the network
- The firewall on the jukebox allows access to port 8880
- The CD player or ripper identifies your server and port (8880) as its CDDB database provider

Caution

You should not attach any server to the Internet until it has been secured. Unless you want to go to the effort of becoming a fully supported CDDB mirror, you should not consider sharing your CDDB database outside of your home or business.

To check if your CDDB database is accessible, type the following from another computer on your network that has the `telnet` command:

```
# telnet localhost 8880
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
201 jukebox.linuxtoys.net CDDBP server v1.4PL0 ready at Fri Nov
29 23:02:46 2002
stat
210 OK, status information follows (until terminating `.')
Server status:
    current proto: 1
    max proto: 5
        .
        .
        .
Database entries: 704884
Database entries by category:
    data: 3944
        .
        .
        .
    classical: 62671
    soundtrack: 22053
```

After the connection was established, I typed the preceding `stat` command. It listed information about settings on the server. It also showed how many database entries there were in each category. To find out what other information you can get from the CDDB daemon, type the **help** command. To exit, type **quit**.

At this point, any CD player or ripper that can use a CDDB database can contact your server to get CDDB information.

To make your CDDB server a bit more useful as a public server, there are a few other things you might want to do:

- **Configure your CDDB server for CGI access** — This lets people look up CDDB information through your Web server. This is useful for them in cases where they are behind a firewall that can use HTTP (port 80) but won't allow CDDB requests (port 8880).

  You need to identify a directory to contain the CGI script needed to make this work (typically you would use `/var/www/cgi-bin`). Then you have to get the cddb.cgi script to put in that directory. The script must be accessible and executable by everyone.

- **Get automatic updates** — To keep your server up to date automatically, you need to make a request to freedb.org to transmit CDDB updates to your server. Otherwise, you will have to download updates yourself periodically.

  If you offer to be an official freedb-mirror site, you can get automatic updates to your database. Otherwise, we would not recommend getting automatic updates from a public server on the Internet. Once you sign up to get automatic updates, you must edit the `/usr/local/share/cddbd/access` file to include a permissions line that appears like the following:

  ```
  host_perms: ches freedb.org connect post update get
  ```

If you aren't able to get these features working, there are other ways of updating your CDDB. You can both add entries manually and get monthly updates of the CDDB (as described in the next section).

Adding CD information to the CDDB

Eventually, you will run into a CD that isn't in the database. The CD may either be very obscure or brand new and not yet in the database. The next section describes how to add an entry to your CDDB database (so you can rip it immediately) and also submit that new entry to Freedb.org. The section after that describes how to add the ongoing updates to your database and rebuild the fuzzy matching database so it can be searched efficiently.

Creating a CDDB database Entry

My brother gave me a CD of holiday music that he and his friends recorded. When I tried to add it to my Christmas mix of music, my jukebox spit it out. So I went through the following steps to add the CD information to my CD database.

Note

You need to have a GUI installed on the jukebox to do this procedure. You also must have the kscd package installed (from CD #3 of the Red Hat Linux installation CDs).

1. Open a Terminal window as root user.
2. Turn off the jukebox software by typing:

```
# service ltjukebox off
```

3. Start the kscd CD player by typing the following:

```
# kscd &
```

4. Insert the CD you want to add to the database.
5. Click the Configure kscd button on the kscd player.
6. Select FREEDB in the left column. A screen appears that lets you add information about your CDDB server.
7. Make the following changes to the kscd Configuration window:
   - Enable auto save to local database. Select this item so that an X appears in the box next to it.
   - freedb Base Directory. Change this directory to the path to your cddbdb directory: `/usr/local/share/cddbdb/`.

   Click Apply to apply your changes and then OK to continue. Figure 3-4 shows an example of the kscd Configuration window.

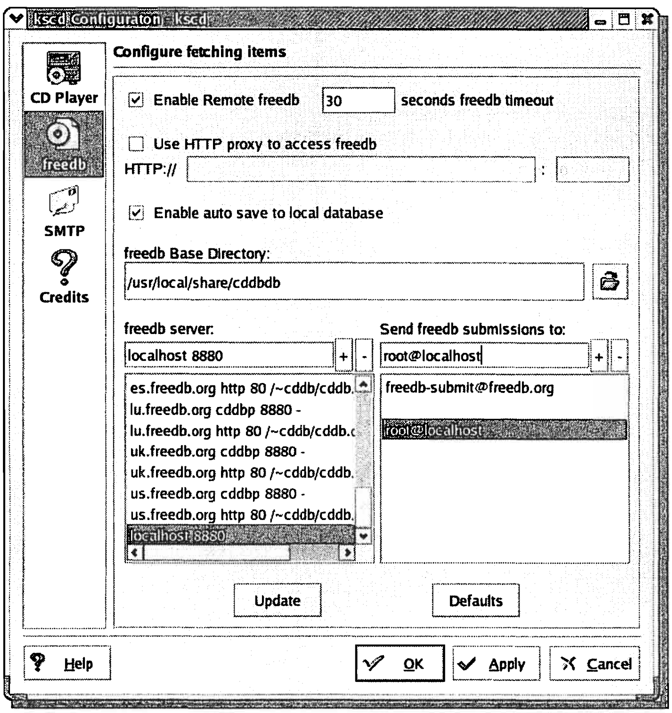


FIGURE 3-4: Set up a CDDB-enabled player, such as kscd, to submit CDDB information.

8. Back at the kscd player, click the Freedb Dialog button. The CD Database Editor window appears.
9. Type the information about the CD: Artist, CD Title, and each track name. Figure 3-5 shows an example of this window.

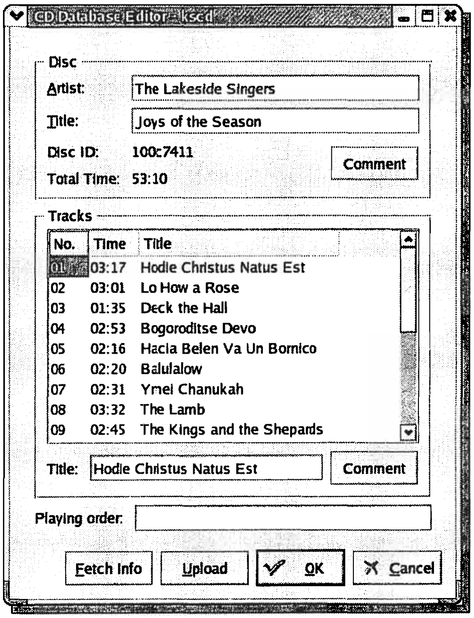


FIGURE 3-5: Add Artist, Title, and Track information to the kscd CD Database Editor.

10. Click Upload to send the CD information (via e-mail) to be added to the master CDDB database (to the address listed on your configuration screen under "Send freedb submissions to"). You are asked to enter the category.
11. Click the category (rock, reggae, classical, and so on) and click OK.
12. Back at the Editor screen, click OK to add the CD information to your local CDDB database (`/usr/local/share/cddbdb/`). You are asked to enter the category.
13. Click the category (rock, reggae, classical, and so on) and click OK.
14. To check that the information was successfully added to your local CDDB database, you can use the `ls` command. For example, using the Disc ID and category in the preceding example, I would type the following:

```
# ls -l /usr/local/share/cddbdb/classical/100c7411
```

At this point, you can restart the jukebox. It should find the database entry you just created.

Rebuilding the CDDB database

You can rebuild the CDDB database after any updates to your local CDDB database to make searches of the database more efficient. Updates can come in the form of:

- Automatic updates (if you get official feeds from the master CDDB).
- Installs of monthly official CDDB updates. To download these updates, go to the downloads section of freedb.org and look for the latest file labeled `freedb-update` (each update file includes the date of the update in its name). Copy the file to `/usr/local/share/cddbdb`. Use `bunzip2` and `tar` commands to uncompress and untar the updates to the proper location.
- Manual updates you do yourself (like the one shown using `kscd`).

Rebuilding the database takes a long time! I usually rebuild it when I don't need the computer for a couple of hours. Here is what you do to rebuild your local CDDB database:

1. Open a Terminal window as root user.
2. Turn off the jukebox software by typing:

   ```
   # service ltjukebox off
   ```

3. Run the `cddbd` command using the `f` (create the fuzzy matching hash file, so the database can guess at inexact matches), `d` (run in debug mode), and `v` (verbose) options, as follows:

   ```
   # /usr/local/bin/cddbd -fdv
   ```

You can restart the jukebox now, and the new CD information should be available.

Changing ripping options

The `cdparanoia` command rips the songs from your CDs with the music jukebox. The `oggenc` command encodes the songs. The result is compressed music in OGG file format that doesn't have a discernable loss of quality.

If you are an audiophile and have a different idea of what "discernible loss of quality" is, you might want to trade some additional disk space for a bit better quality music. You are also free to use different encoding and playback tools. Here are some suggestions on ways you can modify how your jukebox rips and compresses:

- **Change encoding quality**—To change the quality level the `oggenc` uses to encode your music, you need to edit the following line in the `/usr/local/bin/ltautorip` script:

  ```
  my @encoder_opts = qw/-m 4 -q 6 /;
  ```

  The number 6 defines the quality level being used. You can change this number to any number between 0 and 10 (for example, `-q 7`). You aren't limited to integers (a quality of 5.5 is legal to use). You may want to try out a few different levels. Higher numbers result in bigger files and better quality.

- **Change music location** — Say you are running out of disk space on `/usr/local/share/music` and you want to start ripping and playing music to a different directory. You can do this by changing the following line in the `/usr/local/bin/ltautorip` script to some other directory name:

  ```
  my $oggdir = "/usr/local/share/music";
  ```

  After that, you need to change the `/usr/local/bin/musicd` script, replacing `/usr/local/share/music` with the new directory containing your music.

  ```
  /usr/bin/ogg123 -q -z /usr/local/share/music
  ```

- **Use a different encoder** — Instead of using ogg123, you can use any program for encoding music that you prefer. You can change the following three lines to replace the path to the encoding command (`/usr/bin/oggenc`), any options passed to the encoding command (`qw/-m 4 -q 6 /`), and the file extension placed at the end of the encoded files (ogg), respectively:

  ```
  my $encoder = "/usr/bin/oggenc";
  my @encoder_opts = qw/-m 4 -q 6 /;
  my $encoder_ext = "ogg";
  ```

  Next, you need to change the `musicd` script to replace the `ogg123` command that actually plays the music. (The musicd line you have to change is the ogg123 line shown in the previous bullet.)

  ```
  /usr/bin/ogg123 -q -z /usr/local/share/music
  ```

Chuck's Tech Tip

When ripping CDs, it is important to note that the file format and the encoding format are two completely different things. We have chosen to encode the musical data using the Vorbis reference encoder and save it in the Ogg file format. As newer versions of the Vorbis reference encoder are developed, a given encoding quality level will result in better-sounding music that takes up about the same amount of space. A corollary to this is that you'll be able to record at a lower-quality level and achieve the same result, while taking up less disk space.

Troubleshooting Your Jukebox

The Linux Toys Jukebox was intended to run pretty much without any intervention. On occasion, you might want to do some clean-up or deal with a problem CD. The next little sections describe some problems you might have and ways to fix them.

The jukebox spits out your CD

You put in a CD to rip and it immediately ejects. There are a few reasons this might happen:

- The CD is badly damaged.
- The CD isn't a commercial music CD.
- This CD is a music CD, but it isn't in the database.
- The hard disk is full.

To figure out possible problems, you should attach a keyboard and monitor (or log in over the network) and try a few things:

- Try playing the CD to make sure that it contains music and is playable. (Even if it is playable, however, it could still be too scratched up to rip.)
- If you simply have to find a way to add the CD information to your CDDB database, refer to the "Adding CD information to the CDDB" section earlier in this chapter.
- To find out if your hard disk is full, type `df -h`.

Temp files left behind

If you eject a CD or stop the jukebox while a CD is ripping, it will probably leave some files behind in `/tmp`. This is not a big deal, but over time they could add up. After the CD is no longer ripping, you can remove the `/tmp` files as follows:

```
# rm /tmp/tmp*.wav
```

As root user, the `rm` command is run interactively, so you are asked if you want to delete each file that is there (type **y** to delete them). The asterisk causes all files in the `/tmp` directory that begin with `tmp` and ends with `.wav` to be deleted. Be careful how you use asterisks and other wild-card characters! An asterisk by itself will match every file and directory in the chosen directory.

Note

In reality, having too many tmp files isn't a big problem. A feature called tmpwatch runs automatically to delete any files in the `/tmp` directory that have been there for more than 10 days.

Errant processes

If a CD is in the middle of ripping when you turn off the jukebox, `cdparanoia` (the ripping application) may keep going. After you turn off the jukebox, you can check if any `cdparanoia` processes are running. Here's one way to do that:

```
# ps ax |grep cdp
1570 pts/1  D  00:1  /usr/bin/cdparanoia -qw 1 /tmp/tmp.1553.1.wav
```

You can just let that process run its course and then delete the tmp file it creates. Or you can kill the process using the `kill` command with the process ID of the process as an argument. For example, to kill the preceding process, type the following:

```
# kill -9 1570
```

The `-9` will cause all but the most seriously nonuser mode programs to be killed. You can repeat this step by listing and killing `oggenc` processes as well.

Removing music you hate

When you weren't looking, your brother slipped a CD of the Greatest Hits of John Philip Sousa into your CD jukebox. And while you do love marching-band music, it just doesn't go with the contemporary mix you were establishing for your jukebox. (Or you might just be running out of disk space.)

To remove a song or an entire CD from your jukebox, you can use the same `rm` command you used earlier. To remove a whole directory of stuff, however, you can use the `-r` (recursive) and `-f` (force) options. The directory will just disappear, without asking any questions. For example:

```
# rm -rf /usr/local/source/music/John_Philip_Sousa-Greatest_Hits/
```

Caution Be really careful with the `-rf` options to `rm`! A misplaced asterisk or in improper space between `music/` and `John` could cause your whole music collection to be erased! However, it is *convenient* to remove an entire directory, as well as all files and subdirectories under that point.

Checking the CDDB log

Anything the `cddbd` daemon does is logged in the `/usr/local/share/cddbd/server/log` file. By checking the contents of this file, you can see:

- **All requests made for CDDB information by your jukebox.** Each time the jukebox tries to rip a CD, you can see each ID that is queried and the information that the daemon returns. This is useful if you want to see what ID is being requested when a CD is rejected (usually because it is not in the database).
- **Requests from clients over the network for your CDDB database.** If you have a network connection (and opened your firewall to allow connections to port 8880), others can request information from your CDDB database. All those requests are logged in to this file.
- **Database updates.** If you have set up your CDDB to accept automatic updates (with the post option), you can see those updates in this log.
- **Other error conditions.** If there are problems with entries in your CDDB or problems starting the server, they are logged in this file.

The size of the log file is determined by the `log_hiwat` and `log_lowat` parameters in the `/usr/local/share/cddbd/access` file. Information is rotated out once the log file size has reached its limit. You can increase these values to store more log information.

Enhancing the Jukebox

If you and your friends think this project is cool, maybe you have some ideas about how to make it better. The ltjukebox is available from Linuxtoys.net as an open source project, and we

(and we hope you) will be making enhancements to it over time. Here are some ideas we have for possible future enhancements to the Linux Toys Jukebox:

- **CD Images/Web pages** — Create a way to have either an image or a Web page appear in association with the current CD/song.
- **Search and Sort** — Using some database application (such as MySQL), add the CDs to a database so songs can be searched for, sorted, and gathered.

If you have your own ideas, we'd be happy to hear from you.

Understanding the Jukebox

This last section just provides a reference to the location of files and directories that are important to the Linux Toys Jukebox. Most of the commands and files for the Linux Toys Jukebox are contained in the `/usr/share` directory structure. Commands that are particular to this project are contained in the `/usr/local/bin` directory.

Note

When I talk about commands, I sometimes make distinctions between scripts and binaries. With a script, you can open the command in a text editor, see what it does, and (in some cases) change it. A binary command cannot be changed without having the source code and recompiling that code. As a rule, binaries execute more efficiently, and scripts are convenient to view and change at runtime.

Jukebox files and directories

Figure 3-6 illustrates most of the files and directories that make up the jukebox. This view is from the `/usr/local/share` directory.

Here is a breakdown of the contents of your music jukebox:

- **The CD Database Daemon (`/usr/local/share/cddbd`)** — Files in this directory set how the CD Database daemon process (cddbd) behaves. You don't have to change any of the settings in these files for the default jukebox. However, if you want to share your CDDB with others, you can use files in this directory to set information such as who is allowed to access your CDDB server and who can post updates. The server subdirectory contains: the `log` file (which logs all activities of the cddbd daemon) and `fuzzy_index` (which contains the fuzzy matching hash file of all the songs).
- **The CD Database (`/usr/local/share/cddbdb`)** — This directory and its subdirectories contain the hundreds of thousands of CD listings. Each CD is contained in a separate file and is categorized by type of music (blues, classical, country, folk, jazz, and so on). You typically only use this set of directories to make changes and updates to the CD database.
- **Music (`/usr/local/share/music`)** — Each ripped CD is represented by a directory under the music directory. Typically, the directory is named for the artist and CD title, separated by a dash. Words are separated by underscores. Each song is in a separate file, beginning with its order number on the CD and ending with the compression type used (.ogg for Ogg Vorbis).

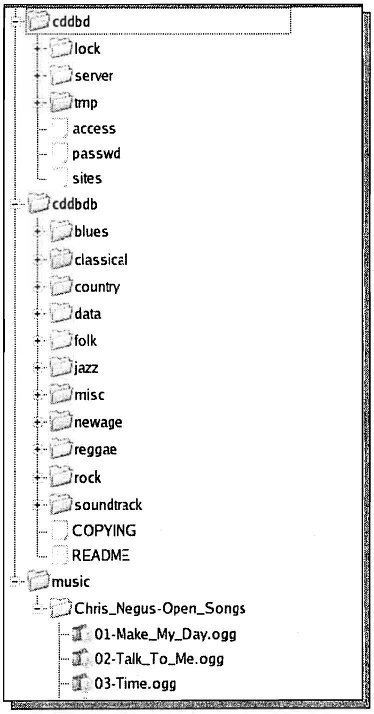


FIGURE 3-6: Linux Toys Jukebox files and directories in /usr/local/share

The player (musicd)

The jukebox start-up script starts the `musicd` script (from `/usr/local/bin/musicd`). It runs the `ogg123` player to play your songs. The `ogg123` player gets your list of songs in one of two ways:

- **All songs** — The `musicd` script runs the `ogg123` command to play all Ogg files (files ending in `.ogg`) from the `/usr/local/share/music` directory. (This is the default behavior.)
- **Playlist** — If the file `/usr/local/share/music/playlist` exists, the `ogg123` player plays the song files that are listed in that `playlist` file. (You have to create the playlist yourself if you want to select your own list of songs from the database you created.)

Note In the "Make your own playlists" section, I describe how to create a playlist to use with the jukebox (instead of just having all of your music play randomly).

The `ogg123` command (as it is configured in your `/usr/local/bin/musicd` script) runs with the `-q` option (to play without outputting any messages to your screen) and `-z` option (to randomize play of the songs you feed it).

The ripper (ltautorip)

The script that starts the CD ripping (/usr/local/bin/ltautorip) does a few more things than just copy the songs from your CD. The ltautorip script is based on the autorip script written by Jonathan Mayer (autorip.sourceforge.net). Once ltjukebox launches the ltautorip script, that script listens continuously for a CD to be inserted into your CD drive. Then it does the following:

- Checks your CD database to see if the CD is a known music CD. (If it is not, the CD is immediately ejected.)
- Uses the cdparanoia command to copy songs (as .wav files) to your /tmp directory
- Uses the oggenc command to compress and store the files into OGG format
- Creates a directory (named for the CD title) and file for each compressed song (named for the order it comes on the CD and song title) in the /tmp directory
- Copies the entire CD directory (containing the compressed songs) to /usr/local/share/music
- Ejects the CD (ready for you to insert your next CD to rip)

The ltautorip script was designed to be simple for bulk ripping of CDs. When you have ripped a bunch of CDs, list the contents of the /usr/local/share/music directory as follows to see what you have (press the space bar to page through the list; q to quit):

```
$ cd /usr/local/share/music
$ ls -CF * |less
Alan_Jones-More_Music:
01-Hopeless.ogg             06-You_re_So_Bold.ogg*
02-Helpless.ogg             07-Go_Get_em.ogg*
03-Can_t_Help_It.ogg        08-All_Around.ogg*
04-Guess_I_m_Useless.ogg    09-My_Day.ogg*
05-If_It_s_Hot.ogg          10-Rain_Will_Glow.ogg*

Bridget_Plumber-Here_We_Are:
01-Big_Dancer.ogg           05-Float_My_Boat.ogg
02-There_We_Go.ogg          06-Where_We_Been.ogg
03-Here_It_Is.ogg           07-Guess_That_s_It.ogg
04-Ain_t_No_More.ogg        08-Well_So_What.ogg
:
```

With your songs in place, there are a lot of different ways you can play them back. There are also ways you can change the ltautorip script itself. See the "Playing with Your Jukebox" and "Modifying Your Jukebox" sections for some ideas of things to do.

The database server daemon (cddbd)

The CD database server daemon (/usr/local/bin/cddbd) basically runs in the background, waiting for requests for information from your CDDB database. It is set up in the jukebox to listen to port number 8880. The cddbd command can also be used to rebuild your CDDB database (described earlier).

To change how the database is accessed (in particular, if you want to receive automatic updates), you can edit the `/usr/local/share/cddbd/access` file. To see log information for your CDDB database, refer to the `/usr/local/share/cddbd/server/log` file.

Summary

Although the Linux Toys Jukebox is intended to rip and play your CD music collection without much help from you, there are plenty of ways to play with it and expand it if you like. You can play music manually (using the `ogg123` command) or from playlists you create. Because the player jukebox contains a full CDDB database, you can even turn your jukebox into a CDDB server that shares artist, album, and song information from CDs with others.

Building a Digital Home Video Archive

chapter 4

I f you have kids (or if your parents have kids), there are probably piles of videotapes lying around of first steps, important games, and school plays. If those tapes are destroyed or stolen, you'll lose priceless images that just can't be replaced. The Linux Toys Digital Home Video Archive project lets you take those precious home videos and:

- Encode them to your computer.
- Play them on your computer.
- Burn them to CD (about 17 minutes) or DVD (up to about two hours) to play back on a computer.
- Burn them to VCD to play them on most home DVD players.
- Pass them around however you like. (For example, you could add video files to your FTP site, as described in Chapter 11, so family and friends can download and view your videos.)

Okay, you won't end up with broadcast-quality television from this project. (And it can be a bit tricky getting it all working.) However, with the right hardware and a bit of tweaking, you can make some reasonable backup copies of irreplaceable family moments. The important thing is that you can then store the originals in a safe and protected place. You can also make copies to distribute to your friends (see Figure 4-1).

in this chapter

- ☑ Choosing hardware
- ☑ Installing Red Hat Linux and Toys software
- ☑ Installing video and television cards
- ☑ Setting up xawtv
- ☑ Recording synchronized audio and video
- ☑ Playing audio/video with mplayer
- ☑ Burning data to CDs and DVDs
- ☑ Making video CDs (VCDs)

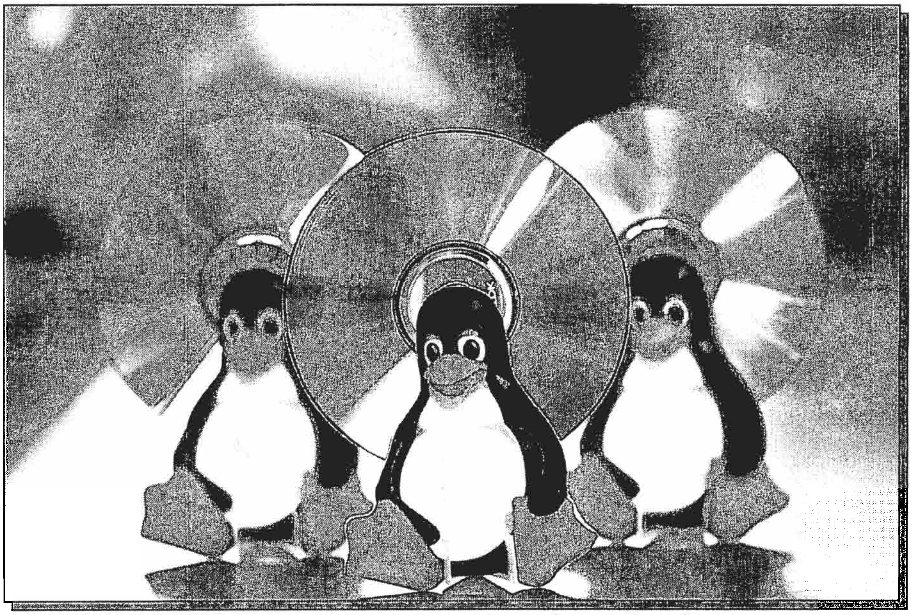


FIGURE 4-1: Burn home movies to CD or DVD and give them to friends.

What's in the Home Video Archive?

Video recording and playback are not quite ready for prime time when it comes to open source software. There are several projects that are about to do some really great things with video. However, they are fragile enough that if you don't get everything exactly right, they will break. This brings us to what we did for the Home Video Archive project.

We put together the Home Video Archive using pieces from several different open source projects. In particular, the video is captured using a combination of tools from the ffmpeg (`ffmpeg.sourceforge.net`) and nvrec projects (`nvrec.sourceforge.net`). To view and adjust your video input, we use the xawtv television player. To play back the video, we use the mplayer video player.

Caution This and the other video projects were among the most difficult ones to get working well. Video projects will not run on minimal hardware! Go carefully through the sections that describe the hardware you need for this project. By getting recommended hardware and installing software as instructed, you'll have the best chance of having success with this project. After that, keeping audio and video in sync and balancing frame rate/size with performance will be your biggest challenges.

Step 1: Gather the hardware

Video is demanding on a computer. The most important considerations when it comes to choosing the computer hardware to use for the project are getting a fast enough processor, a good-quality video card, and a supported TV capture card.

Besides the PC, you need a way to feed the videotapes into your computer. That can be done using several different pieces of hardware, including a VCR or a camcorder.

The personal computer

Although the project may work on slower processors, we are recommending the minimum processor be a Pentium II. We also recommend at least 192MB of RAM for reasonable performance.

CPU and RAM

Video really beats up on the computer's processor and memory. The faster the processor and the better the RAM can keep up, the fewer frames will have to be dropped. If too many frames are dropped, the output begins to look choppy and audio begins to drift.

Table 4-1 shows values we recommend (in the right column) for the basic components of your PC for this project. It also shows the components I used on an extra PC I had lying around the house that had less than the specs we recommend.

Table 4-1 PC Components for Digital Video Archive

Component	*My cheap PC*	*Recommended (for good results)*
CPU	Pentium 3, 650 MHz	Pentium 4, 1 GHz
RAM	384MB	1GB DDR
Hard Disk	40GB	120GB
Size to get reasonably smooth video playback	352 × 240 MPEG-1	1024 × 768 MPEG-1

Red Hat suggests at least 128MB for any graphical install and recommends at least 192MB for good performance. So don't expect any reasonable performance below those numbers.

Besides getting a supported TV capture card (described later), I did add a larger hard disk to my cheap PC. Beyond the requirements to install Red Hat Linux, you need a lot of disk space for storing the compressed video. Otherwise, you will find yourself constantly cleaning up your hard disk so not to run out of space. Space requirements can vary depending on the type of compression, frames per second, frame size, and, of course, how much you want to store on the hard disk.

Note

Chuck, Kevin (our tech editor), and I spent a four-day weekend doing the final builds on the projects in this book. Kevin brought an AMD 400 with 64MB of RAM. The IDE controller was only capable of 16-bit I/O. We popped in a TV capture card and attached some rabbit ears as an antenna. The computer could transmit and play video just fine. Capture had to be done in lower resolution than the default and skipped occasionally. So we could use it to record from a Webcam or TV, but there were too many skips to use it to record VCDs.

Hard disk

Here is the approximate disk space you will need:

- **Red Hat Linux (Personal Desktop install)** — At least 1.7GB of disk space
- **Linux Toys home video packages** — About 100MB of disk space
- **Recorded video** — About 2GB per hour of video you record
- **Making DVDs** — Each DVD can hold up to 4.7GB of data. You are going to need at least twice that amount of disk space to build a disk image to burn to DVD.

I wouldn't suggest starting the project with less than 4GB of disk space. Even with that, you'll have to copy or burn the video off to another medium every time you want to record another video. With at least a 20GB of hard disk space devoted to this project, you won't have to scramble for disk space all the time.

CD-RW or DVD-RW drive

You need a medium for installing Red Hat Linux and Linux Toys software. You also need a way to burn your video onto some removable medium. A CD-RW or DVD-RW drive can serve both purposes.

Given the compression tools and settings we recommend for this project, you should be able to store about one hour of video in about 1.4GB of space. Therefore, on each of these recording media, you should be able to store the following amount of video:

- **CD** — A 700MB CD-R or CD-RW should hold about 30 minutes of video.
- **DVD** — A 4.7GB DVD-R disk should hold more than three hours of video.

If you don't have a DVD recorder, you might want to look into buying one. Prices have gone down drastically from what they were even a year ago. Linux supports burning to DVD. Using DVDs can save you the trouble of splitting up your videos.

Video card

Many low-end PCI video cards won't work well for this project. Most NVidia video cards will work well, provided you download NVidia's proprietary video drivers for Linux.

We took the approach of just trying out the video card or on-board video that comes with the PC. If it didn't perform well, we used an NVidia card and usually solved the problems. Later in this section, I step you through installing and configuring an NVidia card.

TV capture card

For this project, you connect your video input through a TV capture card and use the xawtv player to set properties for the video coming in through that card. Any TV capture card you use for this project must include a chipset that is supported by the video4linux drivers (in particular, we used cards requiring the bttv driver).

There are not a huge number of TV cards supported by video4linux bttv drivers. Capture cards supported by those drivers include the bt848 and bt878 chip sets. Two TV cards that we purchased and got good results from are:

- **Hauppauge WinTV Go** (purchased for about $48 from buy.com). This is a good basic card for playing and recording TV and other video input.
- **Hauppauge WinTV Theater** (purchased for about $138 from buy.com). This card is geared toward connecting your audio and video output into a high-quality entertainment system.

Although Hauppauge is not a household name in the United States, it's easy to find their TV cards in many computer-electronics stores. We recommend these cards simply because we didn't have good luck with some of the other TV cards we tried.

Both of the Hauppauge cards worked well, for this project and the other video projects (Chapters 5 and 8). Both cards have connectors for TV (cable or antenna), audio output, audio input, and S-Video input (where you can connect an external camera). The WinTV Theater card has a few extra features, such as a remote control, more audio output, and FM radio antenna connector (try the `radio` command that comes in the xawtv package to play FM radio).

To find other TV capture cards you can use with this project, you need to install the kernel-source package included with Red Hat Linux. (You can get Red Hat Linux in *Red Hat Linux Bible*, as well as other ways.) Cards that can be used by the bttv driver are listed in the `/usr/src/linux-2.4/Documentation/video4linux/bttv/Cards` file. Be careful to match the model numbers shown, because cards from the same manufacturers are not always supported. If you want to get the latest bttv drivers for TV capture cards, check out `http://bytesex.org/bttv/`.

With a supported TV capture card installed, the next time you boot Red Hat Linux, the card should be automatically detected.

Sound card

You need a supported, configured sound card for this project. See Chapter 3 for information on choosing and configuring the sound card.

Also, you may not be able to record sound directly from your TV capture card. For that reason, you may need a special patch cable (which comes with some TV capture cards) to connect the Line Out on the TV capture card to the Line In on the sound card.

VCR

We wanted to do this project with items that you might already have around the house. We figured that most of you would have a VCR you can use to play the videotapes that you are recording. If you have VHS-C tapes and no adapter, you can play the tapes through a video camera. If you don't have cables needed to plug your camera into the TV card's Video In, you could probably plug it in through your VCR.

Speakers

Any PC speakers will do for this project. You can plug speakers directly into the TV capture card (when you are watching TV input) or to your computer's sound card (when you are playing back video). If you do patch your TV Line out to your sound card Line In, you can leave your speakers attached to your sound card.

Step 2: Install Red Hat Linux

Because you need a GUI for this project, you can begin with a Personal Desktop install of Red Hat Linux. Of course, you can just do an Everything install instead, which will eat up some disk space, but save you from having to pick through the dependencies.

Cross-Reference

See Appendix C for information on installing Red Hat Linux.

You can use either the GNOME or KDE desktop for this project. If you installed both environments (only GNOME is installed by default with a Personal Desktop), a user can select which to use from the graphical login screen. GNOME is used by default.

Step 3: Install Linux Toys software

There are a whole bunch of packages required to encode, play, and produce video and audio for this project. The primary packages are the following (look in Appendix A or on the Linux Toys CD for a more complete list):

- **ffmpeg** — Tools to capture and convert video and audio input
- **nvrec** — Tools to help keep audio and video synchronized during recording
- **mplayer** — A front-end for playing video and audio in a variety of formats
- **vcdimager** — Tools to create VCD video images that can be burned to regular CDs to make VideoCDs

We created an install script to install the entire set of packages needed for this project. To run that script, insert the Linux Toys CD into your CD drive and type the following as root user from a Terminal window:

```
# mount /mnt/cdrom
# cd /mnt/cdrom/ch04-VideoArchive
# ./install.me
```

You are ready to start configuring your Home Video Archive.

Step 4: Install a video card

If you already have a supported video card in your computer, skip to the next step. If you are adding a new video card, here is how to go about adding that card:

1. Before installing the new video card, boot Red Hat Linux and log in.
2. Open a Terminal window and become the root user (type `su -` and enter your password when prompted).
3. Edit the `/etc/inittab` file to change the initdefault to 3. (This will keep you from having scrambled video if your video card is not automatically detected when you reboot.) You will probably change from 5 to 3, so the line appears as follows:

   ```
   id:3:initdefault:
   ```

4. Shut down your computer, unplug the power cord, physically install the new video card (per manufacturer's instructions), and connect your monitor cable to the new card. Then reconnect the power cord.
5. Boot your computer. As Red Hat Linux boots up, it should detect your new video card and ask if you want to configure it. Say that you do want to configure it.
6. When you see the text-based login, log in as the root user.
7. Type the following command to configure your video card:

   ```
   # redhat-config-xfree86
   ```

   If your video card was detected, you should see the Display Settings window, as shown in Figure 4-2.

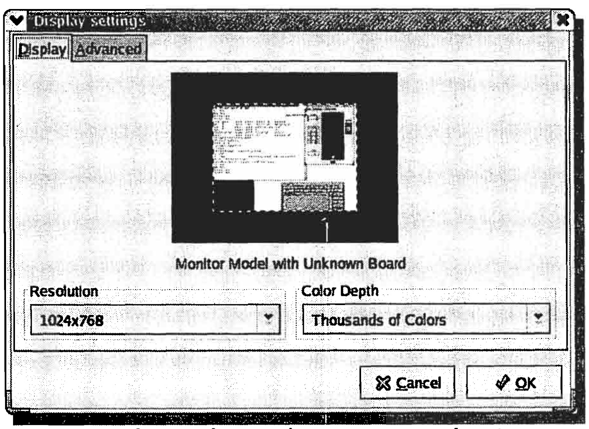


FIGURE 4-2: Change the Display Settings window to configure your video card.

Getting NVidia Drivers

The driver for NVidia cards that comes with Red Hat Linux (the nv driver) doesn't include many of the advanced features needed to support streaming video. NVidia, *however, does* offer proprietary X drivers for its cards that you can download and install in Linux. Although these steps may change if NVidia changes its procedure, here is what I did to get and install a driver for my NVidia card:

1. Go to the NVidia Web site (`www.nvidia.com`) and select Download Drivers.
2. Select Linux & FreeBSD Drivers.
3. Select IA32 (for drivers for the PC).
4. Follow the instructions for downloading and installing the drivers. When I ran it, it had me click the link to `NVIDIA-Linux-x86-1.0-4363.run`. I downloaded it to a directory, changed to that directory, and (as root) typed:

   ```
   # sh NVIDIA-Linux-x86-1.0-4363.run
   ```

5. After the install was complete, I ran `redhat-config-xfree86` again. But it was not able to detect my card properly with the new driver. To create a working configuration file that I could edit further if I needed to, I typed the following (skip the first `cp` for a fresh Red Hat install in which no video was set up):

   ```
   # XFree86 -configure
   # cp /etc/X11/XF86Config /etc/X11/XF86Config.original
   # cp /root/XF86Config.new /etc/X11/XF86Config
   ```

6. Now you can test that X is working by typing:

   ```
   # startx
   ```

7. If the GUI is working at this point, the **`nvidia`** driver should be installed. You can run **`redhat-config-xfree86`** again to adjust the resolution and number of colors. You may have to run **`redhat-config-mouse`** to configure your mouse.

8. You can adjust the following settings from the Display Settings window:
   - **Resolution** — Higher resolutions look better (and let you fit more on a screen), while lower resolutions can perform better.
   - **Color Depth** — Likewise, higher color depths allow more colors, but can slow performance (which can be an important issue when displaying video).
   - **Monitor Type** — The window tries to probe your monitor. If it can't detect the monitor type, you should go to the Advanced tab and configure it yourself. You need to refer to your monitor's documentation for the vertical and horizontal sync rates you need to use.

- **Video Card Type** — Like the monitor, the Display Settings window tries to detect your video card. If it is not properly detected, go to the Advanced tab and select a video card (and its driver) from the list presented.

   When you are done, save the changes and close the window.

9. To check that your video card is working, log out as root, log in as a regular user, and type the following:

   ```
   # startx
   ```

10. In any case, if your GUI is working now, you can also go back to the `/etc/inittab` file and change the initdefault back to 5. If you have an NVidia card, see the sidebar on Getting NVidia drivers to get and install a new driver before you reboot.

Step 5: Install a TV capture card and video equipment

Go through the following steps to install your TV capture card and other video equipment:

1. Shut down your computer, unplug the power cord, and physically install the TV capture card (as described by the manufacturer). Plug the power cord back in when the card is installed.
2. When you reboot, kudzu should detect that the card was added and ask you if you want to configure it. Say that you do.

Note

If for some reason the TV card isn't detected by kudzu, you can try to load the TV card's drivers manually. After the computer finishes booting, as root user from a Terminal window, type `modprobe bttv`.

3. Because we are recording premade videos in this project, you need to connect either your VCR or your video camera to play the videos into your TV card. The VCR can connect directly to the TV In using a standard coax cable. The video camera can be connected to the VCR or, if you have the right cable, directly to the Video In and Audio In jacks on the TV card.
4. To record the sound through the sound card, you may need to connect a patch cable between the TV card's Line Out and the sound card's Line In connectors. (This will allow you to record from the `/dev/dsp` device, as shown in our example a few steps ahead.)

Your hardware should now be set up to record video from your VCR or video camera.

Step 6: Configure the TV player with xawtv

The xawtv TV player lets you play television audio and video in Red Hat Linux. Setting up xawtv has the dual purpose of making your video driver ready to be recorded from and allowing you to watch input from your television or video camera.

To do this procedure, I recommend that you start with your cable TV or TV antennae attached to the VCR or directly to the TV capture card. The following procedure describes how to configure xawtv.

Note

If your video is still completely scrambled after you have followed the instructions for setting up xawtv, refer to the "Troubleshooting Video" section in Chapter 5 for tips on how to get xawtv working.

1. Start Red Hat Linux and log in. (If you don't see a graphical screen when you login, type `startx` after you complete the login.)

2. Start some video playing by either playing a video or by temporarily connecting your television antennae or cable TV cord to your TV card.

3. To start `xawtv`, first turn down the volume on your speakers (to avoid hearing loud static), and then type the following from a Terminal window:

   ```
   $ xawtv&
   ```

   Don't worry if the window is scrambled at first. We'll try to fix that.

4. Position the mouse over the xawtv window and click the right mouse button. The Options window appears, as shown on the left in Figure 4-3.

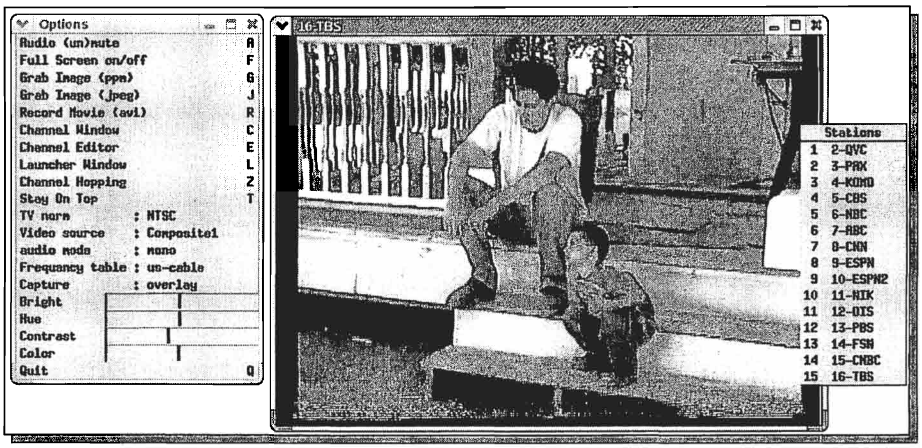


FIGURE 4-3: Set and save your xawtv TV channels and other options.

5. Click TV norm and select your value from the list. In the United States, set TV norm to NTSC.

6. Click Video source and select Television.

7. Click Frequency table and select the input type. For example, in the United states, select us-bcast, us-cable, or us-cable-hrc, depending on how you get your television signal.

8. You should add at least either channel 3 or 4 (depending on which is used by your VCR to play videos). Chapter 5 describes how to add all the channels, along with their station IDs, to your xawtv configuration. Click the main xawtv window and press the up and down arrow keys to select the first channel you want to add.

Note

If you are in the United States, make sure that you set the options described in the first few steps. In Europe, you will use PAL instead of NTSC. If your video is still scrambled after you move the arrow keys up and down for a while, you may need to go to the "Troubleshooting Video" section in Chapter 5 to debug the problem.

9. In the Options window, click Channel Editor. The Config window appears, ready to add information about the current channel.

10. In the Config window, type the station ID for the channel and (optionally) a hot key that will allow you to jump right to that channel. Click Update and Save to update information for each channel. (Click the xawtv window and press the up or down arrow and repeat the procedure for each channel.)

    Right now, you might only want to configure a few channels (in particular, channel 3 or 4, depending on which is used for input from your VCR). When you get to Chapter 5, you may want to set up more channels to do TV recording.

At this point, you should be able to change channels by:

- Clicking the xawtv window to see the station list you created, then clicking the station you want
- Using the up or down arrows to step among all stations available on your television service
- Clicking the wheel on your wheel mouse (if you have one) to step among all stations in your own station list. (This lets you skip over any blank stations along the way.)

Step 7: Prepare to record

When you close the xawtv window, it leaves the video driver in a certain state. All the settings that were active when xawtv exited (the channel, color, brightness, and so on) are still in effect when you start recording. Therefore, I recommend you do the following:

1. Open an xawtv window again (type **xawtv** from a Terminal window).
2. Play a tape from your VCR and make sure you can see it in the xawtv window. You may have to use the up and down arrows on your keyboard to go to channel 3 or 4 (depending on how your television is set).
3. With the tape playing, you should make some adjustments. Again, using the instructions in "Step 6: Configure the TV player with xawtv," adjust the color, brightness, and other values to suit your own eye.
4. When the video appears as you would like it, save the changes on the xawtv Channel Editor (leaving the channel where you were when you watched the video).

Your setup should now be complete. You should now be ready to start recording your videotapes. If anything isn't working, refer to the "Troubleshooting Video" section in Chapter 5.

Recording Your Videos

You can now play videos to your Linux computer through your TV capture card. With that working, you are ready to begin recording your videos to hard disk. Unfortunately, we don't have a pretty GUI for this project. However, we do have some neat open source tools that are coming along nicely in development.

To record video well in Linux, you have to be willing to play a little. While we tried to pick the best open source video tools available for reading, recording, and compressing video, some of the projects are still considered to be in their early stages of development.

We recommend using the nvrec tools for recording the audio and video from your source device. The nvrec project essentially applies wrappers around several well-known video recorders to ensure that sound and video are properly synchronized. We chose the `ffmpegrec` command, due to its incredible versatility and wide range of codecs and file formats to choose from.

Chuck's Tech Tip

One of the biggest problems that plagued this project was keeping audio and video synchronized. After recording more than a few minutes of video, the result resembled one of those old Japanese overdubbed Godzilla movies. A little research has shown that this has been a universal problem since the early days of television. A few performance tweaks mitigated the problem but didn't solve it completely. We were able to permanently solve the problem with Justin Schoeman's nvrec package. Justin's code accurately timestamps the audio and video streams as they come into the machine. The audio is then dynamically *warped* to match the exact video frame that it came in with.

Here's a little procedure for recording synchronized audio and video in Linux.

Caution

Remember, you are about to start recording video using the channel and settings from the xawtv you just closed. So if you forget to select the channel for your VCR (probably channel 3 or 4) or didn't set brightness, colors, and other values the way you like, you just aren't going to get the results you want. If your video input is coming from the composite or other port, you can select that input as well from xawtv. Bottom line—what you see (when you leave xawtv) is what you get.

1. After you have checked that the video output is being displayed, rewind the tape to exactly where you want to start recording.
2. Double-check that the brightness, hue, color and contrast look good to you. (Click the xawtv window and move the sliders on the window that appears to adjust the picture.)
3. Close xawtv. (You can't record with xawtv holding the video device open.)
4. From a Terminal window, type **aumix** to open the sound-mixer screen. You need to select the device that you will be recording from. In my case, I patched the Audio Out from my TV capture card to the Audio In on my sound card. So, on the aumix screen, I clicked the green P in the left column of the Line slider so that it turned into a red R. (This set the audio input of my sound card as the recording device.) Type **S** and **Q** to save the changes and quit.
5. Make a directory to hold your captured video. For example:

```
$ mkdir /usr/local/share/homevideo
```

6. As root user, create an `/etc/nvrec.conf` file using any text editor. This file holds settings to use when you do the actual recording. The contents of the `nvrec.conf` file are simply options, with each option on its own line, that are passed to `ffmpegrec` (or other nvrec commands). Here is an example of some default settings you can start with (and change later as you refine your recording):

```
{ ffmpegrec
-F 1500
-d /dev/dsp
-v /dev/video0
-w 512
-h 384
-ff avi
-ac mp2
-vc mjpeg
-vq 0
-s
}
```

Here's a breakdown of the options we just set in the `nvrec.conf` file:

- **-F *frames***—This option sets the total number of frames that will be grabbed by `ffmpegrec`. By setting this value (which is infinite by default), you effectively tell `ffmpegrec` how long to record. By default, we are recording 24.99 frames per second. So the formula for figuring how many total frames to grab is:

```
minutes of video  x  60 (seconds)  x  24.99 (fps)
```

  So, to record two hours of video, that would be $120 \times 60 \times 24.99$, or about 180,000 frames. The value shown in our example (1500) would record about one minute of video. Instead of setting `-F`, you can just let the recorder play indefinitely. Then just type Ctrl+C to stop recording.

- **-d *device***—This option sets the audio device to use. The first audio device used by OSS (`/dev/dsp`) is set in our example.

- **-v *device***—This option sets the video device to use. The first video device is used in our example (`/dev/video0`).

- **-w *pixels***—This option sets the width of the video being captured. In our example, we used a width of 512 pixels. This and the `-h` option have a major impact on the quality (higher values might result in lower quality) and file size (higher values will result in larger output files).

- **-h *pixels***—This option sets the height of the video being captured. In our example, we used a height of 384 pixels. Like the `-w` option, `-h` can impact quality and file size.

- **-ff *fileformat***—This option sets the file format used. We chose the avi file format, which can be used with a variety of different compression tools (codecs).

- **-ac *codec***—This option sets the codec being used to compress the audio. Our choice here was `mp2`. Other options are `ac3` and `pcm_s16le`.

- **-vc *codec***—This option sets the video codec being used to compress the video. Our choice here was `mjpeg`. Using `mpeg4` gets smaller files of lower quality video.

- **-vq num**—This option sets the fixed queue scale to 0. This effectively turns it off.

Change any of the preceding options to suit your recording. In particular, you want to set your own -F value so that you can configure how long to record.

7. Change to the directory you created for storing your video. For example:

```
$ cd /usr/local/share/homevideo
```

8. Simultaneously, press the Play button on your VCR to play the video you want to record, and run the `ffmpegrec` command line. Here is an example of the `ffmpegrec` command line you might use.

```
$ ffmpegrec -C /etc/nvrec.conf -o party-06152003.avi
```

This command reads the options you just sent in the `/etc/nvrec.conf` file, *starts* grabbing and compressing video based on those options, then outputs (`-o`) the recorded video to the fine name specified (in this case, `party-06152003.avi`).

After the `ffmpegrec` has run for its specified recording time, it will continue to run for a while as it finishes compressing the video. When the recording is all done, you view the recording using the mplayer video player window.

Although we have made the choice to use avi as the file format and mjpeg as the codec, `ffmpegrec` also lets you record using all the file formats and codecs that ffmpeg supports. For a complete list of supported file formats, video codecs, and audio codecs, refer to ffmpeg documentation (`http://ffmpeg.sourceforge.net/ffmpeg-doc.html`).

Chuck's Tech Tip

It's useful to run xawtv every time you record video in order to make tweaks that account for variances in lighting and camera quality. However, after a time you will probably work up a settings profile that works for almost all cases. You can use the `v4lctl` command to set these on the command line or place them in your `/etc/rc.d/rc.local` file to be set at boot time. For example, in order to ensure that the volume is always high enough, I use `/usr/bin/v4lctl volume 85`. The remaining settings can be found in the `v4lctl` man page.

Playing Your Videos

To play video in Linux, the two best choices are MPlayer and xine. We found MPlayer to be a lot more stable, so we are recommending it for playing back your video.

When you installed mplayer, as instructed in this chapter, it should have created a menu entry for launching the player. So you can start the GUI version of MPlayer by either typing `gmplayer` from the command line or from the Red Hat menu by clicking Sound & Video ➔ More Sound and Video Applications ➔ Movie Player. The MPlayer window should appear as shown in Figure 4-4.

With the MPlayer window open, you can try playing back some of your recorded videos. To select a video file for playback, right click the control panel; then click Open ➔ Play file. In our example, when you are prompted for a file to play, select `/usr/local/share/homevideo/party-06152003.avi` (or whatever filename you used). You can use the control panel to play, stop, pause, and go back. The controls are very similar to ones you would find on a music player.

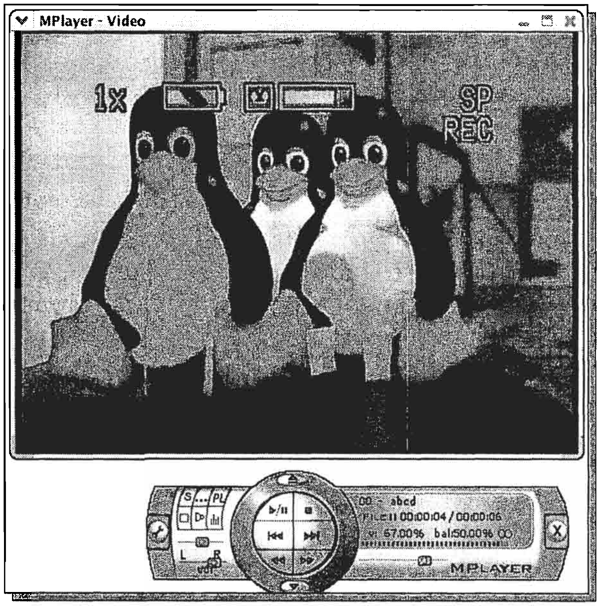


FIGURE 4-4: Play your home videos in MPlayer.

If you want to improve the output of MPlayer, you can try different drivers. Type `gmplayer -vo help` to see what video output driver is being used (X11 by default) and a list of other drivers you can use. Then run gmplayer using a different driver (for example, `gmplayer -vo xv`).

Note

If the video content is not exactly what you would like it to be, you can use a video editor to change the content. Two open source video editing packages to try are Avidemux (`avidemux.sourceforge.net`) and Kino (`kino.schirmacher.de`).

Burning Videos to CD or DVD

Red Hat Linux contains tools for making disk images of our videos or other files and burning them to CDs or DVDs. When you store them in standard ISO9660 format, these disks can be read by any Linux, Windows, or Mac computer. Here is a procedure for taking one or more video files and burning the resulting disk image to CD or DVD.

1. Make the video files into an ISO image using the `mkisofs` command. For example, change to the directory containing the video file or files. Then create the disk image as follows:

```
# cd /usr/local/share/homevideo
# mkisofs -r -o party-06152003.cd party-06152003.avi
 48.69% done, estimate finish Wed Jun  4 02:58:55 2003
 97.44% done, estimate finish Wed Jun  4 02:58:56 2003
```

```
Total translation table size: 0
Total rockridge attributes bytes: 264
Total directory bytes: 0
Path table size(bytes): 10
Max brk space used 4024
10272 extents written (20 Mb)
# file party-20152003.cd
party-20152003.cd:       ISO 9660 CD-ROM filesystem data
```

The result of this command is the party-06152003.cd file. I typed the `file` command to show that it was created as an ISO 9660 CD-ROM file system. The example shows a small 20MB video clip. You can create a file system that is up to 700MB to burn to CD or 4.7GB to burn to DVD. You can add multiple AVI files or directory names to the command line to copy more than one video file.

2. Determine the location of your CD-ROM device by typing the following (as root user from a Terminal window):

```
# cdrecord -scanbus
  0,0,0      0)  "IDE-CD"  "R/RW 4x4x24"  "1.04" Removable CD-ROM
```

In this example, the CD-ROM appeared as SCSI device 0,0,0 (your location may be different).

3. To record the data to CD, you could use the `cdrecord` command with the `-data` option. Change to the directory containing the CD image, and then use the `cdrecord` command as follows:

```
# cd /usr/local/share/homevideo
# cdrecord -v speed=4 dev=0,0,0 -data party-06152003.cd
```

When you run this command, you might be able to increase the speed (or just remove the `speed=4` option to let cdrecord choose an appropriate speed). When the command is done, the prompt will return. You can eject the CD and mark it appropriately. Then you can store it someplace safe or mail it to your Aunt Millie to show her how big your kids have gotten.

Note

If you have a writable DVD drive, use that instead of a CD recorder to record your home videos. In place of the `cdrecord` command, you would use the `dvdrecord` command. Once you determine the device location, you can use the same syntax you did with `cdrecord`. For example:

```
# dvdrecord -v -dao dev=0,0,1 -data bigimage.cd
```

If you prefer GUI tools, the GNOME desktop has a fairly new interface for burning CDs. I've found that it works pretty well when a lot of fancy options aren't required. It also lets you bypass the step where you run `mkisofs` (GNOME does it automatically). These steps describe how to burn CDs using the GNOME Nautilus file manager.

Note If you turned off magicdev as described in Chapter 3, you need to turn it back on for blank CDs to use this procedure.

1. Insert a blank CD into the writeable CD drive. A Nautilus window should open with `burn:///` displayed in the Location box.
2. Open another Nautilus window, for example, by clicking the Home icon on your desktop.
3. In the Location box of the Home window that is displayed, type the name of the folder that contains your videos (no need to create ISO images). For example, `/usr/local/share/homevideos`.
4. Drag-and-drop icons from the folder containing video files to the CD burner folder.

Tip By selecting all the files in the folder, you can see how much space the files you selected need.

5. Click the Write to CD button in the panel. A pop-up window should appear, as shown in Figure 4-5.

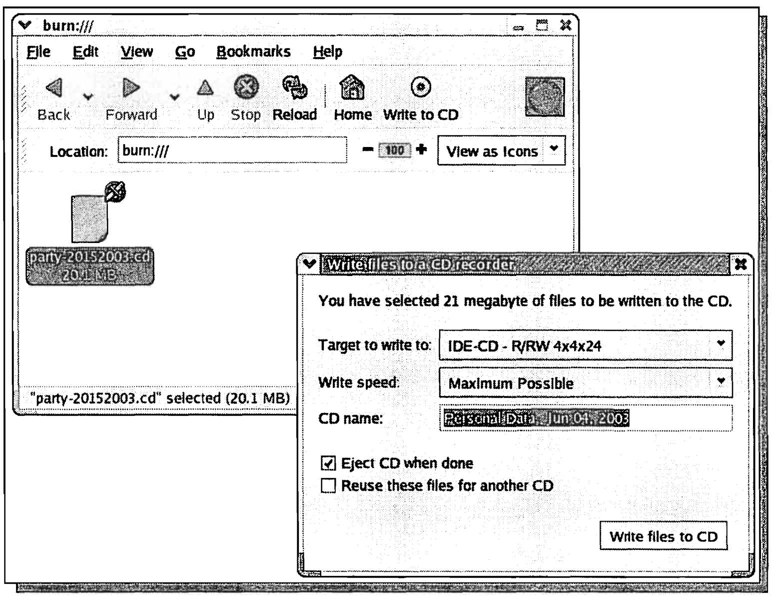


FIGURE 4-5: You can use the GNOME file manager (Nautilus) to burn CDs.

6. You can choose a write speed and the CD player to write to (if you have multiple CD burners). You can also type a label for the CD.

Burning Videos to VCD

The problem with the video you have created so far is that it doesn't play back on the commercial DVD player you have in your entertainment center. Although this technology in Linux isn't quite ready for prime time, the `avi2vcd` script (which was written by our tech editor, Kevin; thank you very much) lets you burn somewhat synchronized audio and video-to-video CDs (VCDs).

Note When you use the `avi2vcd` command to create the .bin and .cue files needed to create the video CDs, you lose the hints that were included in the original avi file to synchronize audio and video. The more frame skips you had during recording, the further your audio and video will drift apart on your VCD. The better the hardware you use to make the recording, the better the resulting VCD will be.

Starting with our example, where you created a video called `party-06152003.avi`, use the following procedure to convert that file into VCD format and burn it to a CD.

1. Change to the directory containing the avi file you want to make into a VCD and run the `avi2vcd` command on it. Here is an example:

   ```
   # cd /usr/local/share/homevideo
   # avi2vcd party-06152003.avi
   ```

   The results of the `avi2vcd` command should be a `party-06152003.bin` and `party-06152003.cue` file.

2. Insert a blank CD into the CD burner on your computer.

3. Determine the location of your CD-ROM device as you did before (type the `cdrecord --scanbus` command).

4. To burn the video to a CD-ROM, you use the `cdrdao` command. Here is an example of the `cdrdao` command (the device and driver for your CD drive may be different):

   ```
   # cdrdao write --device 0,0,0 --driver generic-mmc --eject party-06152003.cue
   ```

Try the VCD in a commercial DVD player (it should work in most of them). Chances are, the video will not fill the screen, playback will be choppy, and audio will be out of sync. Next, mark the VCD "Low quality precious video" and send it to Aunt Millie to play in her DVD player.

Summary

Although video in Linux is still in its infancy, there are some good up-and-coming tools for video available today. We used the `ffmpegrec` tool (which puts a nvrec wrapper around the `ffmpeg` command) to do our video recording and storing.

To set the channel you want to record from, you can run the `xawtv` command. Besides selecting the channel, you can use the xawtv window to adjust color, hue, and brightness.

To play back the recorded video, you can use any player that can play mjpeg files in avi format. We chose the mplayer window, because of its high quality and because it is open source. If you like how the recording turned out, we show you how to burn the video to CD or DVD to play back on your computer. We also describe how to create low-quality VCDs that can be played on most commercial DVD players.

Building a Television Recorder/Player

chapter 5

in this chapter

- ☑ Inside the Linux Toys TV Recorder/Player
- ☑ Setting up television viewing with xawtv
- ☑ Getting TV show listings with XmlTV
- ☑ Viewing listings and recording with WebVCRplus
- ☑ Recording programs with ffmpegrec
- ☑ Troubleshooting video problems

The computer is a natural tool for capturing video. It's true that you need a bit (all right, a lot) more muscle in your computer hardware than you do for other projects. But if you have a PC that's up to the job, you don't need much more than a TV capture card (and the free software we describe) to record and play TV shows from cable TV or an antenna.

The state of free personal video recording (PVR) software that runs in Linux has taken a large leap forward. While several whole projects (such as MythTV) are on the cusp of becoming excellent full-featured PVRs, we decided instead to piece together our own television recorder/player project that lets you:

- Watch TV on your computer.
- Display television listings in your Web browser.
- Select shows you want to record; then record them now or later (automatically) when the shows come on.
- Store and organize recorded shows on your hard disk.
- Play back the shows when you are ready.

This chapter describes the hardware and software you need to create your own Linux Toys Television Recorder/Player. The technology for grabbing and encoding your television programs are the nvrec and ffmpeg packages. Television listings are provided by the XmlTV project. As the front-end to this project, we chose the WebVCRplus interface.

Note WebVCRplus and the other components in the Linux Toys Television Recorder/Player are meant to be used only for private viewing of television in your home. You should respect the rights of those who own the programs you view with this project. In general, you should look at this project as an educational tool and not use it in a commercial way.

What's in the TV Recorder/Player

The primary software packages used in this project include the following:

- **TV player (xawtv)** — Includes the `xawtv` command, which opens a window for viewing live television programs. (This package comes with Red Hat Linux.)
- **TV listings (XmlTV)** — Contains commands used to grab the television-listing information displayed by WebVCRplus from the Internet
- **TV recording setup (WebVCRplus)** — Provides the Web-based interface for selecting the TV shows to record and choosing recording options
- **Video recording (Nvrec)** — Includes the commands (such as `ffmpegrec`) used to record the TV shows and store them on hard disk
- **Video encoding (Avifile)** — Contains the basic commands and codecs we chose to do the recording.

Figure 5-1 shows what the Linux Toys TV Recorder/Player looks like with WebVCRplus displaying television listings and xawtv displaying live television.

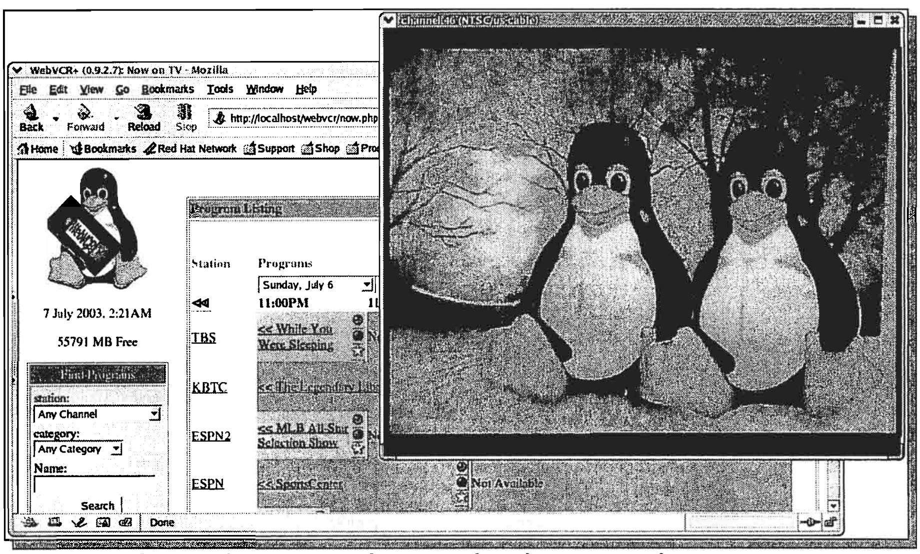


FIGURE 5-1: Select TV shows to record now or when they come on later.

Note

If you really get into using video, you can join the masses in the open source community who are struggling to get video working right (and sometimes, just working at all) in Linux. To dig into some of the details of how the video projects described in this chapter work, start with `/usr/doc/avifile*/README` (once Chapter 5 software is installed). Then perhaps join a mailing list for video4linux, avifiles, nvrec, XmlTV, WebVCRplus, or any other project described in this chapter that interests you.

Setting Up the TV Recorder/Player

The general steps for configuring and using your Linux Toys TV Recorder/Player are as follows:

1. Gather hardware.
2. Install Red Hat Linux and Linux Toys software.
3. Configure TV viewing.
4. Configure TV listings.
5. Add channel listings.
6. Configure TV recording.

Step 1: Gather hardware

This is one project where the real, shall we say, *classic* hardware won't give you very good results. Slower processors and limited system resources will result in missed frame captures and jerky playback. While you can overcome some of that by tuning your video capture, in general, when it comes to video hardware, more is better.

Luckily, hard-disk space has gotten cheap. A 1GHz processor isn't a supercomputer anymore. You can tack a nice TV capture card to your PC for only a few dollars to do almost everything described in this chapter.

Once you have gathered the hardware described here, you need to:

- Install all the hardware (TV card, sound card, video card, and so on).
- Install Red Hat Linux (described next).
- Boot Red Hat Linux. Some hardware will be detected and some will have to be configured to work (described later).

The basic PC

There are a few ways to go when it comes to a PC for this project:

- In the spirit of Linux Toys, get a cheap, used PC and just try to squeeze the best quality video you can out of it.
- Spend a few more dollars (or use the nifty PC you already have) so you can grab video that makes it a pleasure to watch last night's football game (or, as they say in Europe, football match).

Requirements for the computer you use for this project are essentially the same as what you need for the video archive project described in Chapter 4. One difference is that you might want a bigger hard disk.

Because the project relies on both server and Personal Desktop packages, it can be easier to just do an Everything install of Red Hat Linux (requiring about 5GB of space itself) than to sort

through the dependencies. (I discuss the issue further when you go to install Red Hat Linux.) On top of that, you want to figure about 240MB of hard-disk space for every hour of compressed video you want to store.

Chuck's Tech Tip

With RAM more is always better; however, that is not the only thing one should consider when it comes to memory. Most of the work of processing video data is managing the pipeline between the CPU and the hard drive. Data is first loaded from the hard drive to the memory and then steps through the various levels of cache to eventually be processed by the microprocessor (assuming you don't have a hardware video decoder). The best way to improve performance is to either expand the width or speed of that pipeline (or both). Relatively recent advancements in memory have come up with the concept of DDR (Double Data Rate), which increases the speed and width of the memory pipeline. DDR memory is rated in the form PC*XXXX*, where the four *X*s represent how many megabits per second can be run through the memory bus (generally the North Bridge). Make sure your motherboard supports the type of DDR you want to use before you run out and purchase it!

Video card

If you have an AGP slot on your computer, you can use most any AGP video card for this project that is supported by the X Window System in Red Hat Linux. If you have a PCI card, I've found that PCI cards that support Direct Rendering Infrastructure (DRI) work better. (See the DRI project site at `http://dri.sourceforge.net` to find a list of supported cards.)

We tried several different video cards for this project. Many NVidia cards will work (go to the `www.nvidia.com` Web site to download the latest drivers). Matrox G400 or G400-MAX cards will also work well. My cheap PC had an Intel i810 motherboard with an 82810E DC-133 chipset graphics controller (I had to make sure DRI was on).

If you have a supported card, but it is working poorly, there are some ways that you can improve its performance. Refer to the "Troubleshooting Video" section of this chapter for information on improving video performance.

TV capture card

Any TV capture card supported by Video4Linux drivers should work for this project. See Chapter 4 for descriptions of cards that we recommend. We have had good luck with Hauppauge TV cards. A supported TV capture card should be detected the first time you boot your system after Red Hat Linux is installed. (If not, you might be able to get it working by typing the following as root user from a Terminal window: `modprobe bttv`.)

Network card or modem

You need a connection to the Internet (requiring a modem or network card) to be able to download the TV listings to your computer. An always-up Internet connection or one that dials out upon request is preferable, since the TV-show listings are gathered automatically once a week. You can configure an Ethernet network card during Red Hat Linux installation. To configure a modem, refer to Chapter 7, "Creating a Home Network."

Sound card

You probably need a sound card with this project so that you can patch the Audio Out jack on your TV card to the Audio In jack on your sound card. That will let you record the audio portion of your TV shows. See Chapter 3 for information on supported sound cards. (You can configure your sound card the first time you boot Red Hat Linux. Otherwise, use the redhat-config-soundcard window or the `sndconfig` command to configure it.)

Cable TV or antenna

Each TV card has a coaxial connector for input from your TV antennae or cable television. You need to have that cable connected to your TV card when you view or record television.

Step 2: Install Red Hat Linux and Linux Toys software

All software needed for this project is included in either Red Hat Linux or in RPM packages in the `ch05-WebVCR` directory on the Linux Toys CD.

Installing Red Hat Linux

For this project, I recommend an Everything install of Red Hat Linux. As an alternative, you can either begin with a Server installation (then add the GNOME or KDE desktops and xawtv) or a Personal Desktop install. Along with a Personal Desktop install, you need server packages, including Web server (httpd package) and MySQL (all mysql packages) database used by WebVCRplus. You might also have to sort through some perl dependencies (we have included the ones we think you need on the *Linux Toys* CD).

Cross-Reference

See Appendix C for information on how to install Red Hat Linux.

During Red Hat Linux installation, you can configure your Ethernet card (if you have one) and your video card (refer to Chapter 4 on what to do with an NVidia card). The first time you reboot Red Hat Linux, you can configure your sound card. Your TV card will either be automatically detected or you will have to configure it. To see if your Hauppauge TV card was detected (other cards use other modules), log in as root user and type the following:

```
# lsmod |grep bttv
bttv            74179   0 (autoclean)
```

If there is no listing for bttv, try loading it yourself by typing:

```
# modprobe bttv
```

Later, when you try to run xawtv, you can test that the card is working.

Installing Linux Toys software

Next, you need to install the Linux Toys software needed for this project. Insert the Linux Toys CD into your CD drive. First, change to the directory for this chapter and then install the packages as follows:

1. Log in as the root user (or obtain root permission).
2. With the Linux Toys CD inserted, type the following as root user from a Terminal window:

```
# mount /mnt/cdrom
# cd /mnt/cdrom/ch05-WebVCR
# ./install.me
```

Step 3: Configure TV viewing (xawtv)

The xawtv window serves the dual purpose of configuring your video attributes and letting you watch television. You need to make sure that you can receive TV video and audio before you go any further with this project. Here is what you do:

1. Log in as root user.
2. If the GUI doesn't start, start it by typing **startx**.
3. Turn down the sound on your speakers for the moment, so as not to be blasted out by static. (Refer to Chapter 3 for configuring your sound card if it is not working.)
4. Start xawtv by running `xawtv&` from a Terminal window.
5. Configure xawtv channels as described in Chapter 4; then come back here.

Figure 5-2 shows the Options window and Config window that are used to set how xawtv works and to identify the channels that are available.

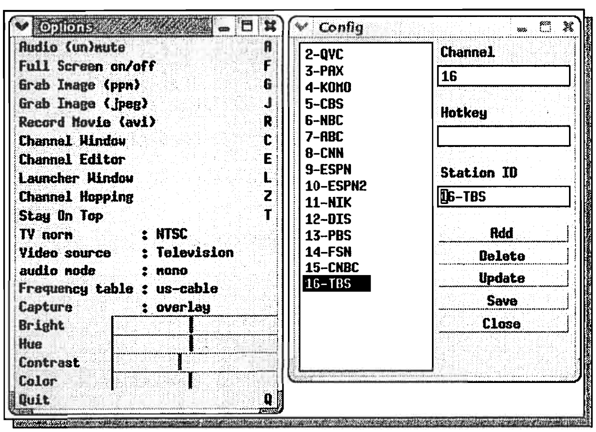


FIGURE 5-2: Tune video and set channels with xawtv Options and Config windows.

You can just add a few channels for now. A bit later, you'll add all the channels you want to be able to view and record using the channel listings you gather from the XmlTV procedure in the next step.

You should now have an xawtv configuration that will work for any of the three video projects in this book (Video Archive, TV Recorder/Player, and Home Broadcast Center). Between projects, you can just use it to watch baseball while you work.

Step 4: Configure TV listings (XmlTV)

To gather TV listings needed with WebVCRplus, we have included XmlTV software. Essentially, with XmlTV you need to:

- **Identify your list of television stations** by your postal zip code and whether you use cable television or an antenna to get TV. Then download the list to your computer.
- **Edit the television stations list** to include only the stations that interest you.
- **Download television-show listings** for your selected stations.

Caution

This is one of the few projects where we're really at the mercy of software that is likely to change over time. If the format of the television show listings used by XmlTV's listing source changes, it will likely break the XmlTV software included with Linux Toys. If the procedure in this section doesn't work, you may need to get an updated XmlTV package from LinuxToys.net or directly from the XmlTV project site at `http://sourceforge.net/projects/xmltv`.

There are commands, beginning with `tv_grab`, that provide the method of gathering TV listings for TV programming around the world. There are programs for grabbing listings for the United States, Canada, United Kingdom, Germany, Austria, Sweden, Norway, Finland, New Zealand, Spain, Italy, and the Netherlands. Here are the commands:

- **tv_grab_na** — Grab television listings for the U.S. or Canada.
- **tv_grab_de** — Grab television listings for Germany or Austria.
- **tv_grab_es** — Grab television listings for Spain.
- **tv_grab_fi** — Grab television listings for Finland.
- **tv_grab_it** — Grab television listings for Italy.
- **tv_grab_nl** — Grab television listings for Holland.
- **tv_grab_nz** — Grab television listings for New Zealand.
- **tv_grab_sn** — Grab television listings for Sweden or Norway.
- **tv_grab_uk** — Grab television listings for the United Kingdom.
- **tv_grab_uk_rt** — Grab television listings for the UK from an alternate source.

For the procedure here, I'm using `tv_grab_na` as an example. If you are in a country outside of the U.S. or Canada, your procedure may vary.

1. Make sure that you are connected to the Internet.
2. Run the following command from a Terminal window while you are logged in as root user (this step takes a few minutes in the middle, so don't worry):

```
# tv_grab_na --configure
Welcome to XMLTV 0.5.10 (tv_grab_na) for Canada and US tv
listings

how many times do you want to retry on www site failures ?
(default=2)
how many seconds do you want to between retries ?
(default=30)
what is your postal/zip code ?94105

getting list of providers for postal/zip code 94105,
be patient..

Choose a service provider:
Select one: [0,1,2,3,4,5,6, (default=0)] [provider #]

you chose 60858 # Comcast - San Francisco - Digital

getting channel list, be patient..
add channel 2 QVC ? [yes,no,all,none (default=yes)] yes
adding channel 3 KWPX...
      .
      .
      .
updating /root/.xmltv/tv_grab_na.conf..
configuration step complete, let the games begin !
```

If you accept the default values for the number of times to retry the site for getting information (2) and number of seconds between retries (30), the first information you enter is your local zip code (in other words, the zip code of the place where your television hookup is). Then, because most places have multiple television services, you need to type the number for the provider you are using (such as your cable, satellite provider, or local broadcast listing).

Note The number after the words "you chose" and before the channel listings begin is important. If you selected the right service provider, this number is one that you can use with other software to view television listings for your television provider.

After that, you are asked (channel-by-channel) to choose which channels to add to your list. You can either:

- Go through the list and either accept (yes) or reject (no) each channel.
- Type **all** to have all channels saved to the configuration file. Then you can edit out the channels you don't want later.

In either case, this command creates the file /root/.xmltv/tv_grab_na.conf file. I recommend editing that file to remove any duplicate or extra channels. Removing channels you don't need saves you time when you download the program listings you need for WebVCRplus. So you may want to remove channels you don't ever watch.

3. Edit the tv_grab_na.conf file with your favorite text editor. For example, type the following from a Terminal window:

   ```
   # gedit /root/.xmltv/tv_grab_na.conf
   ```

   Remove any channels that you don't ever want to see on your TV listings. Keep in mind that you will only be able to record programs from channels on this list. Save the changes and quit when you are done.

4. Because WebVCRplus stores TV listings and other configuration information in a MySQL database, you need to start your mysql server as follows:

   ```
   # service mysqld start
   # chkconfig mysqld on
   ```

5. Because WebVCRplus displays listings and information from a Web browser, you must start the httpd server (as root user from a Terminal window) as follows:

   ```
   # service httpd start
   # chkconfig httpd on
   ```

6. To download and install the television listings for your TV provider, you must run the listings.pl command. Run it from the /var/www/html/webvcrplus directory as follows:

   ```
   # cd /var/www/html/webvcr
   # ./listings.pl
   ```

   You should see a bunch of messages, first about missing stations (for those you edited out), then about TV schedules being downloaded.

   By default, listings.pl will download a week's worth of TV listings for your selected channels. This could take a long time if you are downloading TV listings for lots of stations! After you do this the first time, you don't have to do it again. The information set up in the crontab file will run listings.pl once per week, to keep your listings fresh.

Next, you need to set up your TV channels for recording.

Step 5: Add channel listings

You could have added all your channels by hand when you configured xawtv in "Step 3: Configure TV viewing." But you just downloaded a nice list of channels that match the TV listings in your area when you created the /root/.xmltv/tv_grab_na.conf file. You might as well use that to create your channel list.

When you set up WebVCRplus in the next step, we have the xawtv2 channel selector assigned to do the channel changing when you record. This means that the channels set up in your /root/.xawtv file should match those in your /root/.xmltv/tv_grab_na.conf file. Here's how to do that:

1. Open the `/root/.xawtv` file in any text editor (as root user).
2. Go to the end of the file and insert the contents of the `/root/.xmltv/tv_grab_na.conf` file.
3. Modify `.xawtv` so that there is a separate entry for each channel. Here is an example of what the contents of the `.xawtv` file might end up looking like when you are done, with just the first few channels shown:

```
[global]
ratio = 4:3
freqtab = us-cable
pixsize = 128 x 96
pixcols = 1
jpeg-quality = 75
keypad-ntsc = no
keypad-partial = yes
osd = yes

[defaults]
group = main
norm = NTSC
input = Television
freqtab = us-cable
capture = grab
color = 45%
bright = 45%
hue = 49%
contrast = 40%

[QVC]
channel = 2

[KOMO]
channel = 3
```

The `[global]` section contains general settings that are not associated with particular channels. After that, the `[defaults]` options contain settings that affect all channels. Now comes the fun part.

First, cut out everything from the `tv_grab_na.conf` file that is not a channel listing. Then, for every channel from which you want to be able to record shows, you need to edit the listing you just dumped into this file. Each one needs to be in the form showed for the preceding QVC and KOMO channels.

I admit this is a pain. But you have to do it only once. For each channel, you can also add specific options to better tune the channel. If you want to have the `.xawtv` file you just created applied to every user on your computer, you can copy it to the `/etc/X11/xawtvrc` file.

4. To make sure that the channels were set up right, open the `xawtv` window and click the left mouse button on the screen. You should see the channel list you just created. (Don't worry about the leading numbers. You just want the channel name and number to match those in the `tv_grab_na.conf` file.) Now close `xawtv`.

5. To see if you can change a channel, try running the channel-changing program to make sure that WebVCRplus can record the shows you selected to record. We recommend you use xawtv, with a station name and channel number you choose from your list of channels in .xawtv. Here's how:

   ```
   # cd /var/www/html/webvcr/chanprog
   # ./xawtv CNN 16
   + echo 'Setting station to CNN...'
   Setting station to CNN...
   v4lctl setstation CNN
   ```

   If the station is set successfully, you should be able to start `xawtv` and see the station you just switched to. (If it didn't work, you probably need to correct your station listings in `/root/.xawtv` file.)

You are now ready to set up recording.

Step 6: Configure TV recording (WebVCRplus)

To set up TV recording, we use WebVCRplus (`webvcrplus.sourceforge.net`). WebVCRplus provides a Web-based interface for configuring the recorder, selecting programs, and queuing them to record at the right times. You can also sort and display your TV listings and recorded shows in various ways.

To configure TV recording, open a Web browser and do the following:

1. As root user, open Mozilla or another Web browser and type the following in the location box:

   `http://localhost/webvcr/config.php`

   The WebVCRplus configuration window should appear as shown in Figure 5-3.

2. You can leave these WebVCRplus Configuration options exactly as they are or change them as you see fit and click Save. Here are values you can change:

   - **Directory to store video** — Type the name of a directory where your video can be stored. We recommend the default, `/usr/local/share/television`. This directory should be created automatically by WebVCRplus. To use a different directory, you should make sure that directory exists. For example, to us the directory `/usr/local/share/myvideos`, you could create it from a Terminal window (as root user) by typing `mkdir /usr/local/share/myvideos`. Then change the ownership (`chown apache /usr/local/share/television`).

- **Default number of stations per schedule page** — When you display television listings, by default 20 stations are displayed per page.
- **XmlTV listing source** — Select the name of the `tv_grab` command that produced your channel listings. For example, select `tv_grab_na` and WebVCRplus will know to use the file `/root/.xmltv/tv_grab_na_config`.
- **Minutes to start record before and after** — By default, recording starts two minutes before the show and continues for two minutes after it. (You don't want to miss the credits!)
- **Idle time (minutes) before converting files (0 to disable)** — How long to wait, after recording is done, before compressing recorded video (in minutes)

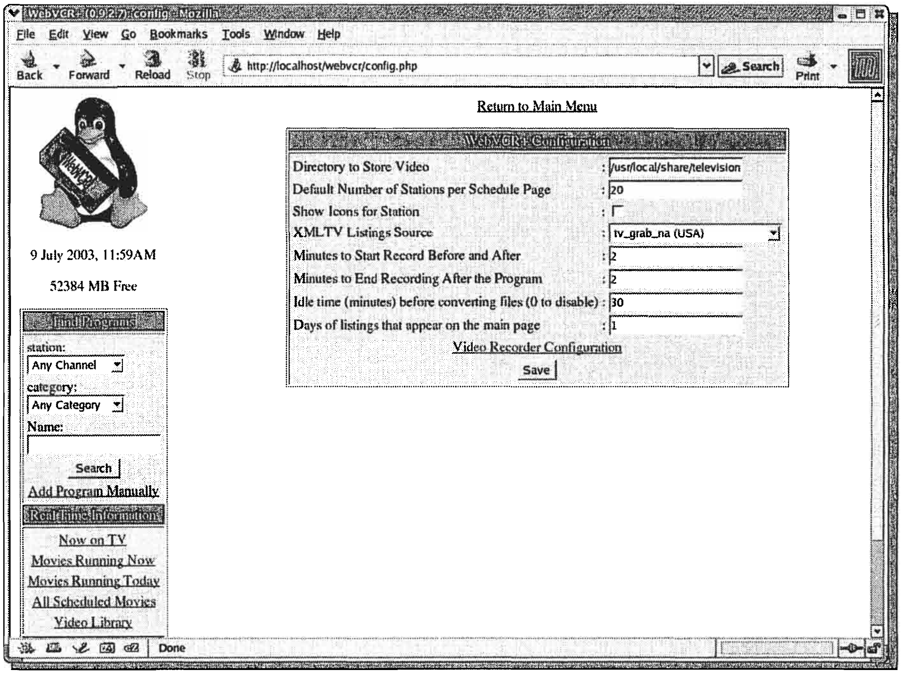


FIGURE 5-3: Use the WebVCRplus Configuration window to set up your TV recorder.

Note In the recommended recording command we use, `ffmpegrec`, we ignore the idle time value and immediately begin compressing the video. So, any change to this value won't mean anything.

- **Days of listings that appear on the main page** — By default, one day of shows selected to be recorded will appear on the main WebVCRplus page.

3. Click Video Recorder Configuration. The Video Recorder Configuration page appears. Change options related to video recording and click Save. The key to configuring this page is the last field: Recording Program. With the default Recording Program Presets set to ffmpegrec, here is how the Recording Program appears (the backslashes indicate that all three lines are actually on one line):

```
/usr/bin/ffmpegrec -d /dev/dsp -s -v /dev/video0 -w 512 -h 384 \
   -vc mjpeg -input #SOURCE# -norm #NORM# -vq #VQUALITY#     \
   -ac mp2 -F $[#TIME#*30] -o #OUTPUT#.avi
```

The recording command is /usr/bin/ffmpegrec. In most cases, this command line will work without modifications. The options to this command set the audio device (-d) to /dev/dsp, the sound to stereo (-s), the video device (-v) to /dev/video, the width of the screen (-w) to 512 pixel and the height (-h) to 384 pixels. The other values on the command line are options that are taken from settings in WebVCRplus window. Here is how the command line is completed:

- **-vc #CODEC#** — The `-vc` option takes the value of the Codec field and translates it into the codec used on the command line. For example, with avi/mjpeg set, when CODEC is replaced, the option looks like `-vc mjpeg`.
- **-input #SOURCE#** — The `-input` option is set to Television by default, from the contents of the Recording Source field. You will leave this for TV recording.
- **-norm #NORM#** — The `-norm` value sets the type of video being received to NTSC from the Norm field. Outside the U.S., you might be using PAL as the type of video.
- **-vq #VQUALITY#** — The `-vq` option sets the video quality from the value in the Recording Quality field. The default is 0 to set the recorder to use variable bit rate for recording.
- **-ac mp2** — The `-ac` option sets the audio output to mp2.
- **-F $[#TIME#*30]** — The `-F` option sets the number of frames that the recorder will record. The `#TIME#` value is replaced by WebVCRplus with the number of seconds that the recording will go. That value is multiplied by 30 (for 30 frames-per-second). The intent is to have WebVCRplus automatically feed the length of time that the record will run.
- **-o #OUTPUT#.avi** — The `-o` option names the output file, based on the television show being recorded. The .avi on the end of the file indicates that the recording is an AVI file.

The other field you can change on this screen is the Channel Changing Program. We assign the `xawtv2` command (described earlier) to WebVCRplus to change the channel before starting to record. It reads channel information from the `/root/.xawtv` file.

4. Before you try to record TV, make sure that your sound card was configured properly. Then type the following, as root user from a Terminal window, to set which audio device to record from:

```
# aumix
```

On the left side of the mixer that appears, you can see the letter P next to each device from which you can record audio, except for one which will have a red R next to it. That device is the one you are set to record from. If you patched your TV card into your audio card as we described in the setup of this chapter, you should click the P next to the Line device so it changes to a red R. Type **S** (to save this change) and **Q** (to quit the mixer). If you later find that you are not recording sound, run `aumix` again and try a different device.

5. To make sure that you are able to record, before trying recording through the WebVCRplus interface, you can try using your recording command line from a Terminal window. Substituting the default values for the variables in the recording command line we give you, you could try out the following as root user from a Terminal window (the backslashes indicate that each line connects to the next; you could remove the backslashes and type it all on one line):

```
# /usr/bin/ffmpegrec -d /dev/dsp -s -v /dev/video0 -w 512 -h 384 \
   -vc mjpeg -input Television -norm NTSC -vq 0                \
   -ac mp2 -F 1000 -o /tmp/test.avi
Starting nvrec.c, version CVS-current (nvrec-CVS-current).
v4l1_core_init(323): Size: 512x384 from /dev/video0
v4l1_core_init(407): Could not get enough buffers - using copy mode
v4l1_core_init(419): Got 4 buffers
v4l1_core_init(421): Emulating 60 buffers
v4l1_core_init(439): Trying video format YUV420
oss_core_init(475): 44100 bps, 16 bits, stereo, from /dev/dsp
oss_core_init(495): 5885 bytes per frame (64,10)

oss_core_init(522): bps=176400, realbps=176400
oss_core_init(530): 689 buffer frags required of 1024 bytes
oss_core_thread(83): Starting!
ffmpegfile_core_init(315): File: /tmp/test.avi
ffmpegfile_core_init(407): Got format : avi (1,1)
ffmpegfile_core_init(426): Video codec id 8
ffmpegfile_core_init(461): Audio codec id 4
Output #1, avi, to '/tmp/test.avi':
  Stream #1.0: Video: mjpeg, 512x384, 29.97 fps, q=3-15, 1800 kb/s
  Stream #1.1: Audio: mp2, 44100 Hz, stereo, 80 kb/s
nvcore_main(446): Initial sync - audio: 1057746359545885012 video
        1057746359540567000 (delta 0.005318 secs)
```

The command just shown tries to record 1000 frames of TV input. The output from this command shows that the command recorded video successfully from `/dev/video0` and audio from `/dev/dsp`. You can see v4l1 starting video, oss starting sound, and ffmpegfile setting the recording file and codecs. Here, the video stream 1.0 is mjpeg and audio stream 1.1 is mp2.

If this command line failed, first make sure you typed every character exactly as shown. Then you might try different video and audio devices, if either of those are not found. (For example, audio might be something like `/dev/dsp0` or `/dev/audio`.) If you get no sound, try `aumix` again from the previous step.

6. If the video looks like it recorded successfully, try playing it as follows:

```
# mplayer /tmp/test.avi
```

If you can play the test file, you are ready to try WebVCRplus to record some TV.

Recording Your Shows

WebVCRplus offers a lot of different ways to record your TV shows. The best way to start is by displaying the TV listings you downloaded earlier. Then you can either:

- **Force Record** — Select the Force Record button to record a TV show that is currently showing. The show starts being recorded immediately.
- **Record** — Select the Record button on a TV show. The show will be queued to start recording when the show is about to come on.

Starting to record

You can begin by displaying your TV listings. From there, you can select which shows to record. Here's how:

1. From a Web browser, type the following in the Location box:

```
http://localhost/webvcr
```

2. Click Now on TV in the left column. Programs currently running for the first 20 channels are displayed. Figure 5-4 shows an example of this window.
3. Here's what you can do once the program listings are displayed:
   - **Program description** — Click the program name to see a description of it. (If you click a movie, a description of the movie appears from the Internet Movie Database, `imdb.com`.)
   - **Force Record (blue button)** — Click the blue button in a TV show box to begin recording the show now. The entry turns red to show that it is being recorded. Recording will stop when the show is over.
   - **Record (red button)** — Click the red button in a TV show box to set recording of the show to begin when the show is ready to come on. (Use Force Record, instead of this button, if the show is already in progress.)
   - **Favorites (star button)** — Click the star button to have the TV show added to a list of favorites. After that, you can click Favorites in the left column to choose to record a favorite show regularly on certain days, times, or channels.

4. Before you try to record, make sure that xawtv or anything else that might have your `/dev/video0` device open is closed. (This is important to remember if you set a show to record in the future as well. If you are watching TV on the screen when the recorder kicks in, recording will fail with a message such as: `open /dev/video0: Device or resource busy`.)

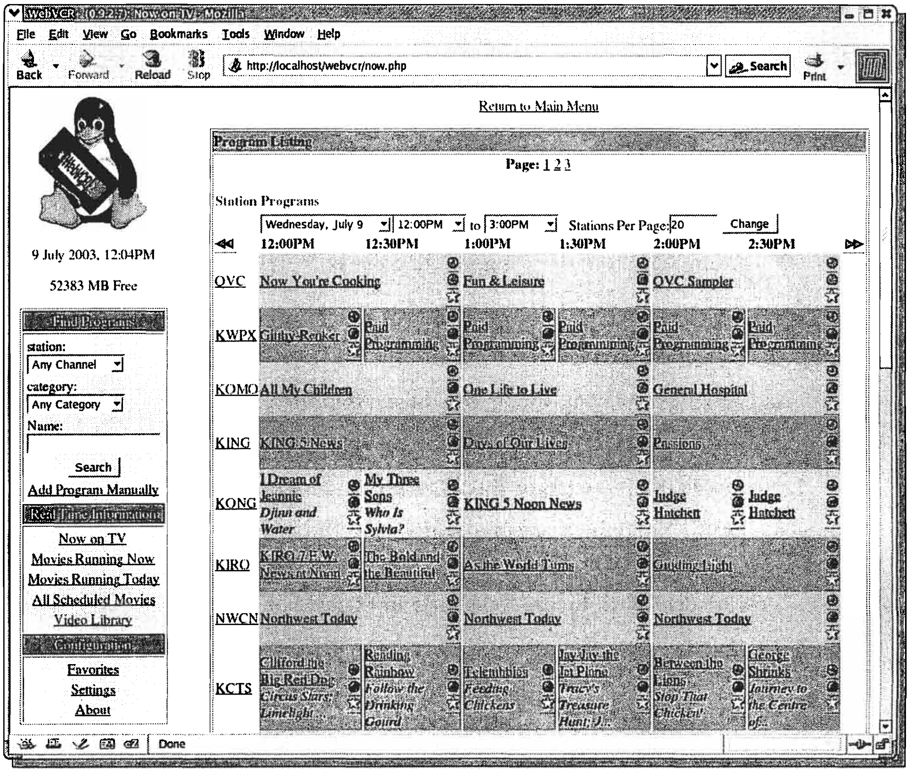


FIGURE 5-4: View current TV listings and select which programs to record.

5. Look for a show that is playing now and click the blue Force Record button on that show's listing. It will take up to a minute before the show starts recording.

6. To check that the recorder is working, open a Terminal window as root user and type the following:

```
# tail -f /usr/local/share/television/data/record.output
```

This command continuously displays information being written to the `record.output` file. Once you begin recording successfully, you should see continuous data about the recording going on. If instead you see error messages, you can use that information to debug your recording command line. (Type **Ctrl-C** to end the `tail` command.)

7. Type the following command:

```
# ls -l /usr/local/share/television/unconverted/*
```

A name representing the program should appear, along with an xml file with descriptive information about the program. The size of the recording should be increasing.

Shows are stored with filenames that include the program name, date recorded, and file format. Here's an example:

```
AskJoe-2003-08-2200-05-00.avi
```

In the previous example, the television show was called Ask Joe. It was recorded on August 22, 2003 (2003-08-22), and was stored as an AVI file (.avi). You can see all the shows you have recorded by selecting Video Library from the left column on the WebVCRplus screen. Click on unconverted to see the files.

Searching for shows to record

In addition to showing the TV shows running at the moment, there are other ways you can have TV shows displayed:

- **Find Programs** — From the Find Programs box in the left column, you can search your TV listings for a particular show (or group of shows). Type a name into the Name box. Then you can limit your search to a certain TV station or program category. Hit Search to start the search.
- **Find Movies** — Click Movies Running Now, Movies Running Today, or All Scheduled Movies to display movies available to record in different ways.
- **Add Program Manually** — Click Add Program Manually to be able to record a channel you select, starting and ending on a particular date and time.

Playing Back Your Shows

To play back your recorded television shows, the Linux Toys CD includes the Mplayer video player. The command used to start the Mplayer video player with a nice control panel is `gmplayer`. To start `gmplayer`, open a Terminal window and type the following:

```
$ gmplayer filename
```

Replace filename with the name of a video file just recorded. For example, a file might be called: `/usr/local/share/television/unconverted/Ask Joe - 2003-08-22 00-05-00.avi`.

Note

Instead of the /usr/local/share/television/unconverted directory, if you had used a recording command that had separate record and compression steps, the resulting video file would appear in the /usr/local/share/television directory.

Use the gmplayer controls to start, stop, skip ahead, or skip back with the video. Figure 5-5 shows an example of the Mplayer window and controls you see when you run gmplayer.

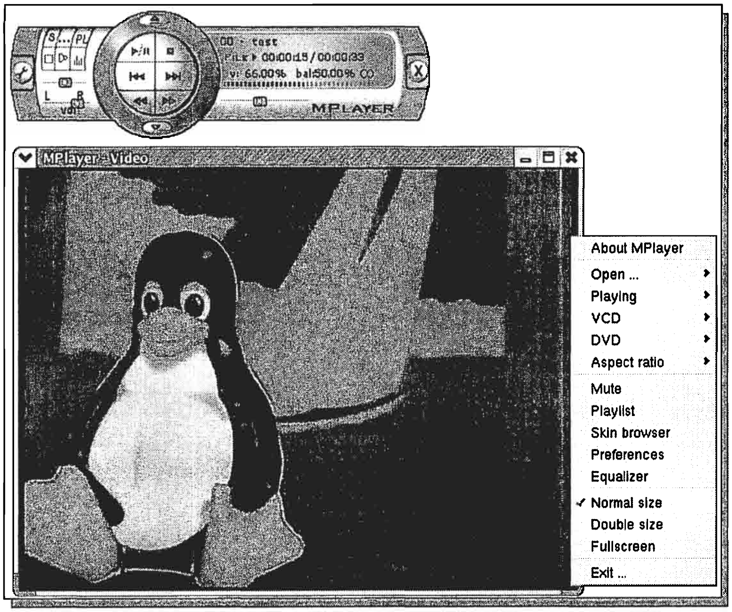


FIGURE 5-5: Use Mplayer controls to play your recorded TV shows.

Remember that the post processing of the video doesn't occur until 30 minutes after the show is done. This is to help prevent the case where you have separate recording and compression programs (which is not the case in our example) and both programs are competing for the processor. If you are in a hurry, you can go immediately to the /usr/local/share/television/unconverted directory and try playing the video file directly from gmplayer.

Troubleshooting Video

Video recording in Linux is still in its infancy. While there are tools for working with video in Linux, you have to do some digging around when things go wrong. Here are a few tips on how to debug video-recording and playback problems you may be having.

- Make sure the video, TV capture, and sound cards are all installed properly. Remember to patch your TV Line Out to sound card Line In before you record. Be sure to connect your television input to your TV card and speakers to your sound card.

- Video cards from NVidia require special drivers to work properly, but most supported cards should be detected properly. A supported TV capture card should be automatically detected. If it's not, you may need to load the driver manually (type `modprobe bttw` as root user). For sound cards, use the `redhat-config-soundcard` or `sndconfig` commands.
- Before you record video, make sure that nothing has the video drivers open. In particular, close any xawtv windows that are viewing live television. Recording will fail if another program has the video driver (`/dev/video0`) open.
- Each time you open xawtv and you change channels, resize the window or adjust brightness, you change the state of the video drivers. Type **`v21-info | less`** to step through the current settings. Try that command before and after opening xawtv and resizing the window or adjusting settings, to see how you change the state of the driver.
- Your particular configuration may be using different devices for audio and video than we describe here. In other versions of Linux, `/dev/video` is used instead of `/dev/video0` for grabbing video. Audio devices may be `/dev/dsp`, `/dev/dsp0`, or `/dev/audio`. Try changing those devices when your record if you are not getting output.
- If you can play audio, but you are unable to record it, run the `aumix` program. Along the left side of the screen, make sure there is a red R next to the device from which you need to record audio (probably Line).
- Try recording from the command line if WebVCRplus is unable to record.
- It is normal to see some frames being dropped when you are recording using the tools we describe. If this becomes excessive, however, video will appear choppy. The nvrec tools we use do a good job keeping audio and video in sync, even if your processor is having trouble keeping up. To get the best results, however, you can try shutting down the GUI and any demanding processes before you begin recording video.
- Remember, this book is called *Linux Toys*. You are not getting state-of-the-art video from the software we describe here. It should be fun to play with and, with some work, should result in serviceable video of TV shows or home movies. Then maybe someday you'll be able to tell your grandchildren that you used WebVCRplus in the days before it ran on a wrist watch and could record 300 television shows simultaneously.

Summary

If you can get a supported video card and TV capture card, the software that comes with the Linux Toys Television Recorder/Player lets you use a Web-based interface to find and select the shows you want to record. Using the Mplayer video player, you can play back your TV shows whenever it is convenient for you.

Creating an Arcade Game Player

chapter 6

in this chapter

☑ Installing Arcade Game Player software

☑ Configuring Xmame, sound card, and joystick

☑ Configuring gRustibus to play MAME games

☑ Getting game ROMs (legally) to play in Xmame

☑ Adapting the PC to an arcade cabinet

Would you love to revisit the days of yore when video games were kinder and gentler? Maybe you still have all the old game cartridges you used to play back in the 1980s in a box at your parents' house. Perhaps the console player is broken or missing? Wouldn't you like to play some of those old games again?

Well, you can. The Linux Toys Arcade Game Player lets you play classic console games from the 1970s–1990s. But instead of running them on an old console, you can play them from your Linux based PC. The Arcade Game Player project is based on some simple yet powerful components, including:

- **MAME** — The MAME project (which stands for Multiple Arcade Machine Emulator) contains a set of console game emulators for playing those old games.
- **Xmame** — The Xmame project consists of a version of MAME that runs in the X Window System (the most popular graphical user interface for Linux and other UNIX systems).
- **gRustibus** — gRustibus is a front-end to Xmame that lets you configure, manage, and launch MAME console games.

There are literally thousands of classic console games that work with MAME. Although some games have a few quirks with sound and graphics, many of the games run as they did originally on your home console or at the arcade.

Although you can simply play MAME games on your Linux PC, a lot of people have had fun adapting a MAME player to work in an arcade cabinet. We also started our own project to adapt an existing game cabinet to work with MAME. However, if all you have at the moment is a working Red Hat Linux system, you can play games in MAME just fine from your PC (see Figure 6-1).

Use this chapter to get Xmame and the gRustibus front end to Xmame running under Red Hat Linux. If you are adventurous, and want to adapt Xmame to your own arcade game console, I'll give you some pointers on how to do that.

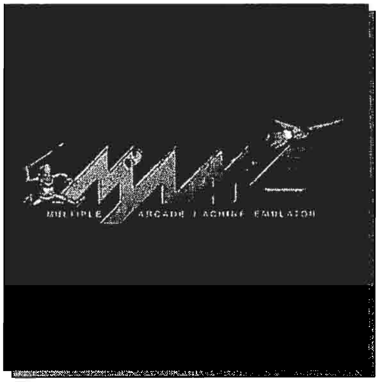


FIGURE 6-1: Play classic arcade games in Linux using MAME.

About MAME Console Gaming

The MAME project began as a way to preserve the workings of some of the old console game players of the 1970s, 80s, and 90s. Being able to play the console games was not the primary objective. As they say at the official MAME site (`www.mame.net`):

> . . . the main purpose of the project is to document the hardware (and software) of the arcade games. There are already many dead arcade boards, whose function has been brought to life in MAME. Being able to play the games is just a nice side-effect.

Our Linux Toys Arcade Game Player is based on that side-effect.

Since Nicola Salmoria began the project in 1996, more than 100 MAME developers have gotten more than 3000 classic console games running in MAME. These games were created originally between 1975 and 2000, with the bulk of the games arising between the mid-1980s and early 1990s.

Games are divided into categories, such as ball and paddle, breakout, climbing, driving, fighter, maze, multi-play, pinball, puzzle, quiz, shooter, sports, table top, and wrestling. Games you might remember include:

- Donkey Kong
- Pac-Man
- Asteroids
- Air Wolf
- Space Invaders
- Q*bert

And there are a few thousand other games available as well. Some of the more popular games are also available in multiple MAME versions.

You can control MAME games by using the keyboard or mouse or by adding a joystick or Town Pad. You can adjust the size of the game window, frame rates, and scale.

The biggest issue related to playing MAME games is the question of obtaining the game ROMs legally. That issue is discussed in the "Getting Game ROMs" section.

Configuring an Arcade Game Player

We've divided this project into several major parts. The first part ("Configuring an Arcade Game Player") describes how to get the MAME games running on a regular PC along with Red Hat Linux. The other parts tell how to configure and play the games. The general steps for configuring the software you need for your arcade game are as follows:

1. Gather hardware
2. Install Linux and Arcade software
3. Set up the joystick

Step 1: Gather hardware

The PC you need for the Linux Toys Arcade Game Player is about the same as the one you would need to run a desktop Red Hat Linux system. On top of RAM, CPU, and disk-space requirements (see Appendix B for details), we recommend having a sound card and joystick.

The personal computer

Because you need a GUI for this project, we recommend doing a Personal Desktop install of Red Hat Linux. Also, the GNOME desktop (which is the default anyway) is recommended because the front-end we are using for this project (gRustibus) requires that the GNOME libraries be installed. Here are some estimates to help you understand disk-space requirements:

- **Red Hat Linux** — A Personal Desktop install requires 1.7GB of disk space.
- **Linux Toys Arcade Game software** — Xmame and gRustibus require about 6MB of disk space.
- **Game ROMs** — There are games with ROMs as small as 32K. Larger games can be more than 8MB each. If you leave 100MB of disk space, you should have no trouble fitting as many ROMs as you own on your computer.

You can probably get by with the minimum amount of RAM (128MB) allowed for a Red Hat Linux desktop system. The recommended amount of RAM for a graphical install, however, is 192MB. A 400Mhz Pentium-class processor is the minimum recommended CPU.

Video card

Most any video card that can run X11 should work with Xmame. Since X is the basic interface for the Red Hat Linux GUI, if the GUI is running you should be able to run Xmame.

Sound card

Most supported sound cards that contain a joystick port will work for this project. Chuck and I both used inexpensive soundblaster cards that work well. See Chapter 3 for information on configuring a sound card in Red Hat Linux.

Joystick

Most any joystick that can connect to the joystick port on your sound card will work for this project. I've been using a Logitech Wingman joystick that my wife, Sheree, picked up at Goodwill for $1.99 (can that girl shop, or what?). Most games work fine with a stick that goes four directions and has two buttons (for shooting, jumping, or whatever the game does).

Chuck's Tech Tip

You might be tempted to use a keyboard instead of a joystick. This will work just fine for a single-player game. When you get to multiplayer games, the keyboard has some limitations. A keyboard is a "polled" device. This means that the computer asks the keyboard many times per second if a key has been pressed. If two users hit a key at the same time, only one of the keystrokes will make it through to the computer. An example of that situation happens in certain games that allow you to hold down a button for extended periods of time, perhaps to shoot a gun. If your opponent hits a button while you are holding down the fire button, your ship will stop firing, and control will revert to the other player until you lift your finger and repress the fire button. This can be very disruptive and might allow one player to have an unfair advantage.

Step 2: Install Linux and arcade software

As we said earlier, a Personal Desktop type install of Red Hat Linux should work for this project. If you want to put together the Linux installation yourself, make sure that you have at least the following Red Hat Linux software installed:

- The X Window System
- GNOME desktop environment (actually, only the GNOME libraries are required)
- Joystick (for configuring the joystick)

The Linux Toys Arcade Game Console software is contained on the Linux Toys CD. To install the software, open a Terminal window as root user and type the following commands:

```
# mount /mnt/cdrom
# /mnt/cdrom/ch06-Arcade/install.me
```

You should see indications that the software packages are being installed.

Step 3: Set up the joystick

Assuming your sound card is already working (if it's not, refer to the sound card configuration section in Chapter 3), you should configure your joystick to work. Here's how you do that:

1. Plug the joystick into the joystick port in the back of your sound card.
2. Make sure that the sound and joystick modules are loaded. As root user from a Terminal window, type the following:

Note

The last line indicates a Soundblaster (es1371) sound card. Replace *es1371* with the module for your sound card. You can use the Red Hat Soundcard Detection (`redhat-config-soundcard`) window to determine the module used for your sound card.

```
# modprobe gameport
# modprobe joydev
# modprobe analog
# modprobe es1371 joystick=0x200
```

The commands just shown will make your joystick work immediately. To make those changes permanent, add the preceding lines (minus the # signs) to the `/etc/rc.local` file.

3. Test the joystick with the `jstest` command from a Terminal window as root user:

```
# jstest /dev/js0
Joystick (Analog 3-axis 4-button joystick) has 3 axes and 4 buttons. Driver
version is 2.1.0.
Testing ... (interrupt to exit)
Axes:  0:    0  1:    0   2:    0  Buttons   0:off  1:off  2:off  3:off
```

Adjust the wheels on your joystick so that each axis lands on zero (0). Click each button to make sure that it works (the word `off` turns to on as each button is pushed). Press Ctrl-C to quit `jstest`.

Getting Game ROMs

The stickiest part of MAME is legally obtaining the games to play with it. The games that you play in MAME are referred to as ROMs. ROMs (which stands for read-only memory) represent the games that were extracted from game cartridges and put in a form that can be played on a computer by MAME.

Even though the games that run in MAME are no longer being sold by the companies that created them, those companies still maintain copyrights to those games. That's why:

- Nobody distributes commercial game ROMs with MAME.
- You should not pass around copies of game ROMs.

There are three games in the public domain that you can use with MAME. Those three games are: Gridlee, Robby Roto, and Poly-Play. We include those three games when you install the Linux Toys version of Xmame and gRustibus.

Owning games

There was a time when the common belief about obtaining game ROMs was that if you owned the game cartridge, it was legal to download a ROM for that game from the Internet. According to the MAME FAQ (`www.mame.net/mamefaq.html`), that is not true. Here is what it says:

You have three legal options:

(1) Search for auctions (on eBay for example) of the actual arcade circuit boards, and buy a ROM reader device to copy the ROM chips' contents into files.

(2) Buy the HotRod Joystick control panel. It is bundled with a CD that includes MAME, the Arcade@Home frontend, and selected 100% legal Capcom ROMs.

(3) There are also three free ROM sets available right on the official MAME site, in the miscellaneous downloads page.

The illegal option is to search the net with Google, Altavista, Yahoo, Webcrawler or other search engine, for the ROM files. Be aware that this is breaking the laws of almost every country. Before you consider doing this, see if the particular arcade games' copyright-owner has the ROMs available (as with Capcom). That way you will support the companies that support emulation.

Using the gRustibus window, you can see what games are currently ready to run in MAME. Figure 6-2 shows the games list that appears in the gRustibus window, with any games that are installed appearing in a different color (in this figure, the first three games are installed).

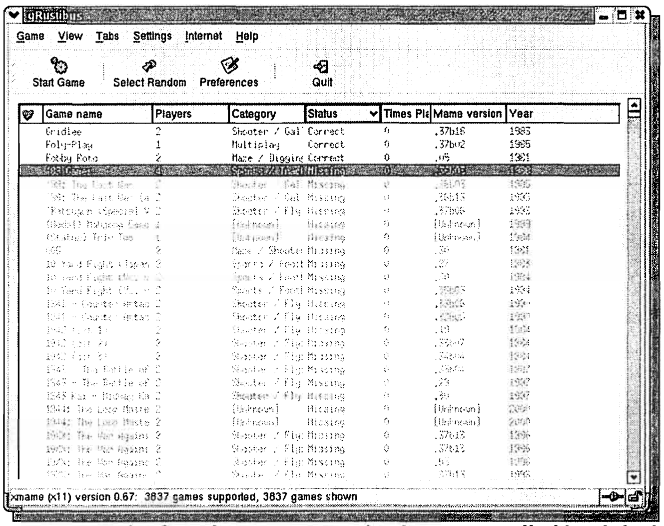
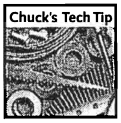


FIGURE 6-2: The first three games in this list are installed by default.

Because the copyright issues associated with MAME ROMs require careful handling, I strongly recommend that you do your own research into the legality of obtaining game ROMs.

Chuck's Tech Tip



The MAME package itself comes with a comprehensive list of every ROM in existence. The primary reason for this is to prevent rogue ROMs from messing with your system. Upon running MAME, the program is able to look at the ROM files you wish to run and determine if they are legitimate or not.

Installing game ROMs

When you obtain game ROMs, you typically get them in a zip archive (.zip file). The name of that zip file is the name that identifies the game. You can use that name to create the ROM directory and to ask for the game if you run it from an `xmame` command line. The following simple steps describe how to install a fictitious ROM called `a.zip` so that it can be used by xmame:

1. Copy the zip file to the ROM directory. For example, here's how to copy it from your home directory:

   ```
   $ cp $HOME/a.zip /usr/share/xmame/roms
   ```

2. Try the `xmame` command to see if the ROMs work. This is a good way to just check that the game will start up and, if it doesn't, it's a good way to see the error messages that are incurred. For example:

   ```
   $ xmame a
   ```

   If this command doesn't work, check that:

   - The `xmame` command is in your PATH (or type the full PATH).
   - The game's ROM directory is in the base ROM directory. (Set the ROM directory with the rompath value in your `$HOME/.xmame/xmamerc` file. Note the dot in the `.xmame` directory name.)
   - The game's zip filename and the game name passed to `xmame` are the same.
   - The case for the ROM files is correct. When the command fails, it will list the proper case for each ROM (probably lowercase and numbers).

Tuning Up the Arcade

We chose gRustibus (`grustibus.sourceforge.net`) as the front-end software for configuring and launching the games. The first time you run gRustibus, you should do some configuration to make it work in a way that suits you.

Starting up gRustibus

The following procedure will help you get gRustibus working initially.

1. Install at least one game ROM as described in the "Installing game ROMs" section. (Or use one of the three games that are installed with this project.)

2. While you are logged in to Red Hat Linux as the user for which you installed Xmame, open a Terminal window and type the following:

   ```
   $ grustibus&
   ```

   The reason for starting it from a Terminal window the first time is so that you can see any problems it may have starting up. If all goes well, you should see the gRustibus window as shown in Figure 6-3.

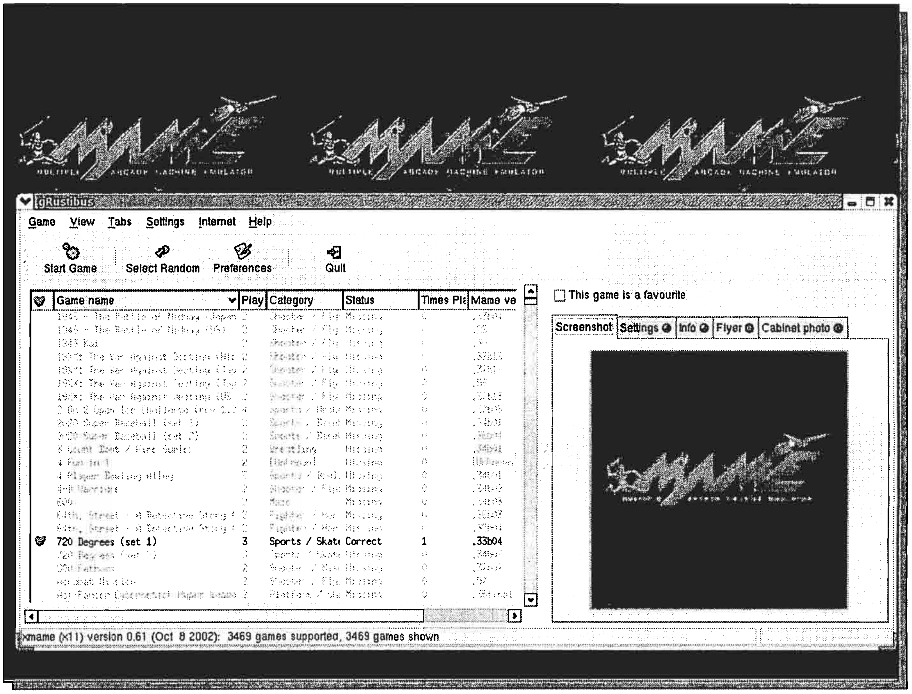


FIGURE 6-3: Use gRustibus to configure and start MAME games.

3. If gRustibus warns you to build your games list, click OK. Then click Game ➔ Rebuild gamelist. All the ROMs available to Xmame should appear on the playlist in the left column.

4. Click the Status column so that the arrow on that column points down. Scroll to the top of the games list, and you should see the word `Correct` in that column next to the game or games you installed. If none of the games are correct, either the game wasn't installed properly (go back and install a ROM properly) or the path to your games is set incorrectly (go to the next step).

5. Click the Preferences button on the gRustibus window and select the Path settings tab. The paths shown should be set as follows:

   - **xmame executable** — Should be set to `/usr/bin/xmame`
   - **All other paths** — Should all begin with `/usr/share/xmame`, followed by the directory or file for the particular feature

   The paths should be set properly. If they are not, you can change them, apply changes, and rebuild the games list. You need to know these locations because those are the places where you put the ROMs, screen shots, photos, or cabinets you want to add to be used by gRustibus and Xmame.

6. Click the Status column again to make sure that your ROMs were found and Correct.

Setting gRustibus preferences

There are a few things you can do to tailor the general operation of gRustibus to your liking. From the gRustibus window, do the following:

1. **Change View** — Click View to see settings that determine which ROMs are displayed. When you start out, you can see listings for all 3000-plus games. Once you have your own games installed, you can remove the checkmarks next to settings that show games you don't have installed, possibly leaving only the following:
   - **Show correct roms** — To see only ROMs that are ready to play
   - **Show favorites only** — To see only games marked as `favorites` (with a heart)
2. **Add favorites** — Click a game that is a favorite of yours. Then select the This game is a favorite check box in the right column. A heart will appear next to the game, so that you can sort to display only favorites or to display them first.

Note

When you first install gRustibus, games that are marked with a heart icon are ones that were favorites from the Rustibus arcade hall, where the developer of gRustibus used to play as a kid.

3. **Add Internet settings** — Click the Preferences. From the Preferences window that appears, click the Internet settings tab. From this tab you can enter information about a Web or FTP site that contains the following:
   - **ROM URL** — Enter a directory on the Web that contains ROM files (in the form game.zip), followed by `%s.zip`.
   - **Screenshot URL** — Enter a directory on the Web that contains screen shots of each game, followed by `%s.png`.
   - **Flyer URL** — Enter a directory on the Web that contains an image of a flyer representing each game, followed by `%s.png`.
   - **Cabinet URL** — Enter a directory on the Web that contains images of arcade cabinets related to each game, followed by `%s.png`.

   In each case, the `%s` will be replaced by the name of the game you select to download. After you have entered these settings, you can download any games that you own legally from the sites you entered as follows:
   - Right-click the game you want to download.
   - Select to download the selected ROM, screenshot, flyer, or cabinet picture.

   The selected game (or related image) should be installed properly for use after the next time you rebuild the playlist.

Setting Up the Games

The artistry of using Xmame is getting the settings just right for each game. The Settings tab on the gRustibus window lets you set permanent values for each game, so you don't have to remember what options you used each time you run the game.

Note The settings you add to gRustibus are given as arguments to the `xmame` command line when you launch a game. If you started gRustibus from a Terminal window, look at the contents of that window after you launch a game. It will show you the options that were used from your settings.

The procedures in this section describe the settings you can change to get your games working at their best.

1. Click a game that you are ready to configure (I assume the game is already installed). Settings for that game should appear in the right column, as shown in Figure 6-4.

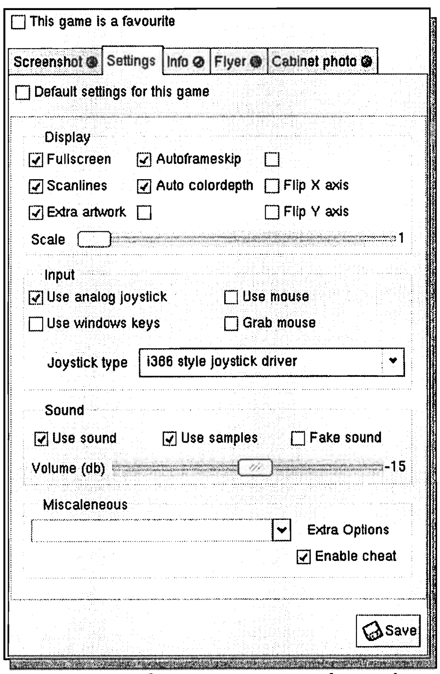


FIGURE 6-4: Define your settings for each game in gRustibus.

2. Click the Settings tab and uncheck the Default settings for this game box. Change any of the options listed in Table 6-1.

Table 6-1 Change xmame game options in gRustibus

Game option	Description
This game is a favourite	Select this checkbox to add a heart icon next to the game in the game list. You can sort your game list so that all favorites appear at the top of the list.
Default settings for this game	Click to unmark this checkbox to be able to change any of these settings.
Fullscreen	Select to run the game in full-screen mode.
Autoframeskip	Tells MAME to skip frames automatically when needed to speed up emulation. This works well on some games that do complex graphics rendering. Other games, such as Crusin, Police Trainer, and games from Incredible Technologies and Williams, run better with this off.
Scanlines	Add scan lines to the screen to make the games look more like they would on old display screens (sort of like adding static if you played an old radio show).
Auto colordepth	Automatically change the color depth of the screen to match the game you are playing.
Flip X axis	Select this if you want to make the game play backwards (from left to right).
Flip Y axis	Select this if you want to make the game play upside down.
Extra artwork	Select this to use extra artwork if it is available (such as extra backdrop and overlay artwork).
Scale	Change the scale of the game.
Input	Choose the input device: analog joystick, use windows keys, use mouse, or grab mouse. If you are using a joystick designed for the PC, select i386 style joystick driver as the joystick type.
Sound	Typically, you will select use sound and use samples to have sound used for the games. If your computer doesn't support sound, but the game requires it, use fake sound.
Volume (db)	Set how loud the sound is in decibels.
Miscellaneous	Add your own MAME command line options. To see what options are available, type `xmame -help | less`.
Enable cheat	Click this checkbox on to enable cheats.

3. Click Save to save the changes to the options for the game.

At this point, you should be ready to start playing the games.

Chuck's Tech Tip

One of the really cool aspects of MAME is that it allows you to turn on and off game cheats. I was able to talk to Tom Griffith, a lifelong gaming expert who filled me in on the history of console cheats. Over the course of time, console debuggers were developed that enabled gamers to peek into the actual contents of the game's running memory and make changes. Certain memory values, when changed, would do things like give you invincibility or advanced weaponry. These memory values were shared among the gaming community, thus enabling people with a little bit of technical expertise and a debugging tool to enjoy aspects of the game that were previously unavailable to them. The MAME group put the values of these "cheats" into a database that is released with the MAME package itself.

Playing the Games

To launch a game, either double-click it or select it and click Start Game. If the game doesn't start, look in the Terminal window from which you started gRustibus to see if there are any error messages that might be useful for debugging the problem.

Figure 6-5 shows the Robby Roto game. This is one of very few games available to use with MAME that can be distributed for noncommercial use. To find out more about getting game ROMs and other issues about MAME, visit Mame World (`www.mameworld.net/`).

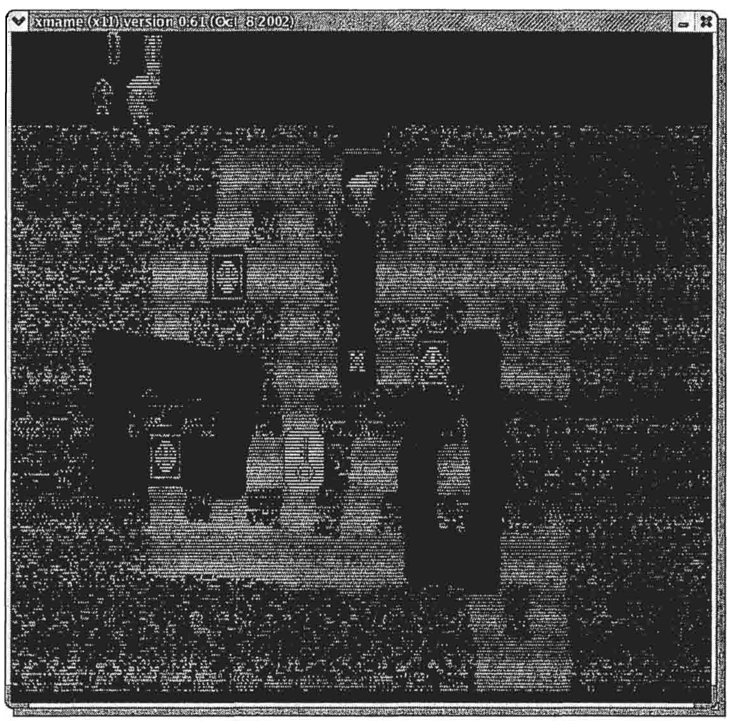


FIGURE 6-5: Robby Roto is freely distributable for noncommercial use.

There are some standard controls that will work with most MAME games. To see how controls are set for a particular game, press the Tab key. After you press Tab, a menu appears that lets you see and change settings for the game. Press the arrow keys to select input. Then you can see which keys perform each function (and change them, if you like).

Table 6-2 shows keys that are typically used with many games. Table 6-3 shows keys that are used to resize the Xmame window.

Table 6-2 Starting, stopping, and displaying controls

Keys	*Result*	*Keys*	*Result*
1	Select one player	2	Select two players
5	Insert a quarter	6	Insert two (or more) quarters
P	Pause	Tab	See and change specific game settings (press Tab again to return to the game)
Esc	Return to the game (when viewing menus) or quit (during game)	F3	Reset the game
F8 / F9	Reduce or increase frame skip	F10	Speed throttling (toggle on and off)
F11	Show speed display (toggle on and off)	Shift+F10	Idle sleeping (toggle on and off)
Shift+F11	Profiler display (toggle on and off)		

Table 6-3 Resizing the game

Keys	*Result*	*Keys*	*Result*
Left-Shift+Insert	Increase widthscale	left-Shift+Delete	Decrease widthscale
Left-Shift+Home	Increase heightscale	left-Shift+End	Decrease heightscale
Left-Shift+PageUp	Increase scale	left-Shift+PageDown	Decrease scale
Left-Alt+Insert	Normal window	left-Alt+Home	Fullscreen mode (DGA)

Actions such as moving, shooting, and jumping can be handled by your joystick. You can also use keyboard keys to control the game. For example, you can use the four arrow keys to move up, down, right, and left. You can use the Ctrl and Alt keys to shoot or jump.

Running Xmame as Arcade Games

With Xmame running happily on your computer, you may want to make that machine into a dedicated arcade game player. There are a lot of ways you could turn your PC into a dedicated Xmame Arcade Player. The following procedure will result in a computer that lets you:

- Boot up the PC to an Xmame login screen
- Log in under a game name (such as `robby` or `pacman`)
- Have the game start immediately in full screen mode
- Exit back to the login screen when you are done

Here's the procedure:

Note

Before you begin the procedure, create an `xmame` command line for each game you want to run from your arcade console. One way to do this is to run `grustibus` from a Terminal window and tune the game. Once it's tuned, run the game from `grustibus` and either cut-and-paste or copy the `xmame` command line that appears in the Terminal window for the game to use later in this procedure.

1. As root user from a Terminal window, add the following lines to your `/etc/inittab` file:

   ```
   # Run Xmame login in runlevel 4
   mm:4:respawn:/usr/X11R6/bin/xdm -nodaemon
   ```

2. Copy the Xmame logo to the xdm directory:

   ```
   # cd /etc/X11/xdm/pixmaps
   # cp /usr/share/pixmaps/grustibus/defaultsnap.png .
   # convert defaultsnap.png defaultsnap.xpm
   ```

3. Edit the `/etc/X11/xdm/Xresources` file so that the `logoFileName` value is set to use the `defaultsnap.xpm` file. This puts the Xmame logo on the login screen. Here is an example of the line:

   ```
   xlogin*logoFileName: /etc/X11/xdm/pixmaps/defaultsnap.xpm
   ```

4. Create a user name that represents the game. For example, I'll create one for Robby Roto called `robby`. As root user from a Terminal window, type:

   ```
   # useradd robby
   # passwd robby
   New password: *****
   Retype new password: *****
   ```

   You can use the name of the game as the password or any password you like.

5. Log out and log in as the new user name.

6. Open a Terminal window and type: **switchdesk**
7. From the Switchdesk window, switch the desktop to TWM and click OK.
8. Open the `$HOME/.Xclients-default` file using any text editor; remove the `exec twm` line and add the command line for `xmame`. It should include all the tuning options you got from tuning it in gRustibus. Here's an example of a line you might add:

   ```
   exec /usr/bin/xmame -fullscreen -rompath /usr/share/xmame/roms robby
   ```

9. Log out and log in again as the new game user. The game should execute immediately and exit back to the Xmame login screen when you are done. (If you have a text login, type `startx` after you log in.)
10. To make sure that the new run level starts up your Xmame Arcade screen, save any data on your current window. Then, as root user in a Terminal window, type:

    ```
    init 4
    ```

11. If the login screen comes up with the Xmame logo shown, try logging in as a game login to make sure it is working.
12. Open the `/etc/inittab` file and change the `initdefault` line value from 5 to 4. When you reboot, the Xmame login screen should come up.
13. Finally, you may want to change the host name to something like Xmame_Arcade in the `/etc/sysconfig/network` file, so it appears on the login screen.

You should now have an Xmame Arcade machine that lets you fire up any game you have configured by simply logging in to that game's login.

Building an Arcade Game Cabinet

As I mentioned earlier in this chapter, Chuck and I went out and bought an arcade cabinet to adapt for use with Xmame. This will be an ongoing project with us. You can check the `LinuxToys.net` Web site to see our progress. If you are impatient, here are a few links to Web sites where people have successfully built or adapted an arcade cabinet for MAME:

- Build a MAME Cabinet in 24 Hours: `http://rtcw.no-ip.org/cabinet/part1.shtml`
- LuSiD's Arcade Flashback: `http://home.earthlink.net/%7Eseanhat/arcade/arcade.html`
- Raza's MAME/Jamma Arcade Cabinet: `www.russprince.com/cabinet/`
- Plain Mame Cabinet: `http://plainmame.emugaming.com/index.html`

Here are a few other tips to help you if you decide to create your own cabinet:

- Some games have a virtual dip-switch on them that lets the game be flopped vertically between players one and two. You can use this feature for a cocktail table–style game cabinet.
- Multiplayer games needing more than one joystick can be hard to configure. Many games, however, need only four cardinal directions and one or two buttons to fire. For this situation, you can buy or build a keyboard adapter. For more information, and a place to buy buttons and digital joysticks, check out the Build Your Own Arcade Controls site (`www.arcadecontrols.com`).

Summary

Those console games you thought you would never see again have gotten new life thanks to the MAME project. More than 3,000 games run in the MAME console game emulator. Using the Xmame version of MAME, you can run those old games on a PC with Red Hat Linux or other Linux and UNIX systems. To launch games with the Linux Toys Arcade Game player, we rely on MAME, Xmame, and a front-end called gRustibus for launching the games.

Projects for the Home

part III

in this part

Creating a Home Network

chapter 7

in this chapter

- ☑ Gathering and connecting hardware
- ☑ Setting up DHCP
- ☑ Creating a firewall
- ☑ Routing
- ☑ Setting up Samba file/print sharing
- ☑ Making the LAN toy-ready

A standalone Linux Toy is cool. A networked Linux Toy is bliss. If you are like me, you have a few computers around your house that are begging to be networked together. And once you start building Linux Toys, it's only going to get worse (or do I mean, better?).

A Linux system can make a great focal point for your home network, letting you:

- Share an Internet connection
- Share printers
- Share files

A home network can be a great asset for your Linux Toys projects as well. For projects such as the Music Jukebox and Digital Picture Frame, which can run without monitors or keyboards, you can use your home network to change playlists or add new images. With an Internet connection, your toy-car races and home-security cameras can be viewed wherever you choose via the Internet.

Of course, you can share any resources you want using your Linux Toys Home Network server. For our purposes, however, we're going to focus on:

- Setting up hardware:
  - Getting the hardware
  - Connecting equipment
- Setting up the server:
  - Configuring a firewall/router
  - Setting up DHCP
  - Using Samba file and printer sharing

The home network that you end up with can help make your whole family a little more digital and a little bit cooler. (See Figure 7-1.)

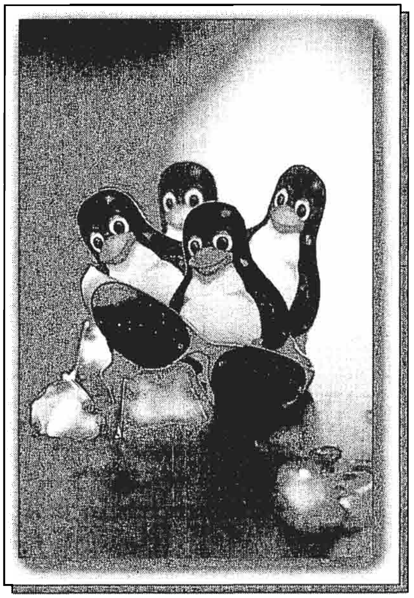


FIGURE 7-1: Your neighbors will wish they had a Linux Toys Home Network server.

Note

There are many other types of services you can set up on your Red Hat Linux Home Network server. Our approach is to minimize the risk of connecting your home computers to the Internet, while maximizing the sharing of basic resources (files and printers in particular) between your home computers. Chapters 11 and 12 describe how to add dial-in connections and more public services to a Red Hat Linux server.

Putting Together the Hardware

The hardware you need for a home network can be as simple as an inexpensive switch or hub and some cables. The computers themselves each need only one Ethernet card (except for the server, which needs two). You can get the stuff you need at any store that sells computer supplies.

Step 1: Gather the hardware

You need one computer to build this project. That computer will be your Linux Toys Home Network server. The following sections describe the components of that computer, as well as the other materials you need to set up your network.

The Home Network server PC

Most any computer that will run Red Hat Linux and has enough slots for two Ethernet cards (or has one or both Ethernet ports built-in) can be used as the Linux Toys Home Network server. Particular components of the PC are described as follows.

Hard disk

Here is the approximate disk space you will need:

- **Red Hat Linux** — We recommend a Server install, with either the GNOME or KDE desktop packages added. This requires at least 1.2GB of disk space. You might ask to see a detailed package list during installation for this project, so you can pick and choose which tools and services you want on your Home Network server.
- **File server** — Because this is the only computer expected to be turned on in the house all the time, we expect that it could be used as the repository for the family shared files. While regular text files don't take up much room, family photos, music, and especially videos can take up a lot of disk space. So the amount of disk space you need varies from a few megabytes to multiple gigabytes of space.

CD drive or floppy disk drive

CDs are the preferred medium for installing Red Hat Linux and the Linux Toys software. However, if you are using an old, floppy-only computer, you can install Red Hat Linux over the network.

Network cards

Two network interface cards are needed in the Linux Toys Home Network server. You probably need only one Ethernet card on each client computer in your LAN. If you are using a dial-up connection to the Internet (not recommended), you need only one Ethernet card and one modem on the server.

Cable modem or DSL

Get a high-speed (DSL or cable modem) service to your house if possible. The ISP will give you the information you need to configure it. Ultimately, you will connect the hardware that your ISP recommends into the second Ethernet card (eth1) on your Linux Toys Home Network computer.

Hubs and switches

In the most basic case, you can get an inexpensive 10/100 Mbps hub or switch. A hub will allow you to connect together your client computers and the server at speeds of up to 100 Mbps (depending on the speed the Ethernet cards support). Unless you paid big bucks for your Internet connection, even 10Mbps on your LAN would be good enough to send out as much data as your Internet connection can handle (and more).

A new five-port 10/100 Mbps hub can be found for under $30. Low-end switches, which have more intelligence and enable you to segment your LAN to improve performance, are available now for only a few dollars more.

Chuck's Tech Tip

A *hub* broadcasts all data to all nodes connected to it. As the number of nodes on a network grows, broadcasting all data to all nodes begins to saturate the network, and no useful work can be done. With a *switch*, all Ethernet nodes contain a (supposedly) universally unique identifier called a Media Access Control (MAC) address. When an Ethernet frame enters a switch, the switch can make a reasonable determination as to which port to send that frame based on the MAC address information presented by the frame. As a result of falling network-equipment prices, it is almost always better to purchase a switch rather than a hub, even for a small home network.

Wiring

For the cables connecting your computers to your hub or switch, you want twisted-pair wiring that meets or exceeds Category 5e wiring specifications. The cable contains four pairs of twisted copper wiring. The connectors on the ends are RJ-45 connectors.

You can purchase premade cables at any computer store or on the Internet. If you're doing long runs throughout your house, however, you will want to run the cables and add the connectors yourself. As an alternative, you can offer your buddy who works at the phone company a case of beer to come over for an afternoon to get you started.

Here are a few things you should know about the computer-network wiring:

- **Length** — Individual runs of Cat5e cable should not be longer than 100 meters.
- **Drop wires to one location** — If your computers are not all in one room, you should try to have your network wires all dropped to one location, if possible. A good technique is to set up the hub or switch in the basement or a large closet and connect your wires all in one place.

  If you put your Linux Toys Home Network server in the same spot, you won't have to listen to the server fan running all the time in your family room when the other computers aren't in use. If you are building a new home, you may also want to drop coax (RG-6) for cable TV and security cameras, wires for whole-house audio, and Category 5e cable (for network and phone).

  Figure 7-2 shows a nice little basement corner for a person whose new house is ready for his Linux Toys Home Network server.
- **Use switches to segment** — Because switches and hubs are so close in price these days, there's hardly any reason to have hubs anymore. There are some cases where you want to use a switch instead of a hub for sure. A switch segments the communications on your LAN, so that when computers on the same segment talk together, computers on other segments of the LAN don't even see the communication. With a hub, all network traffic on the hub can slow down all traffic on the hub. So you might want to put a switch in to segment high-use computers that are in proximity to each other.

You can purchase Category 5e cabling in spools of 500 to 1000 feet. You can purchase boxes of RJ-45 connectors as well, from computer stores, electrical-supply stores, or home centers.

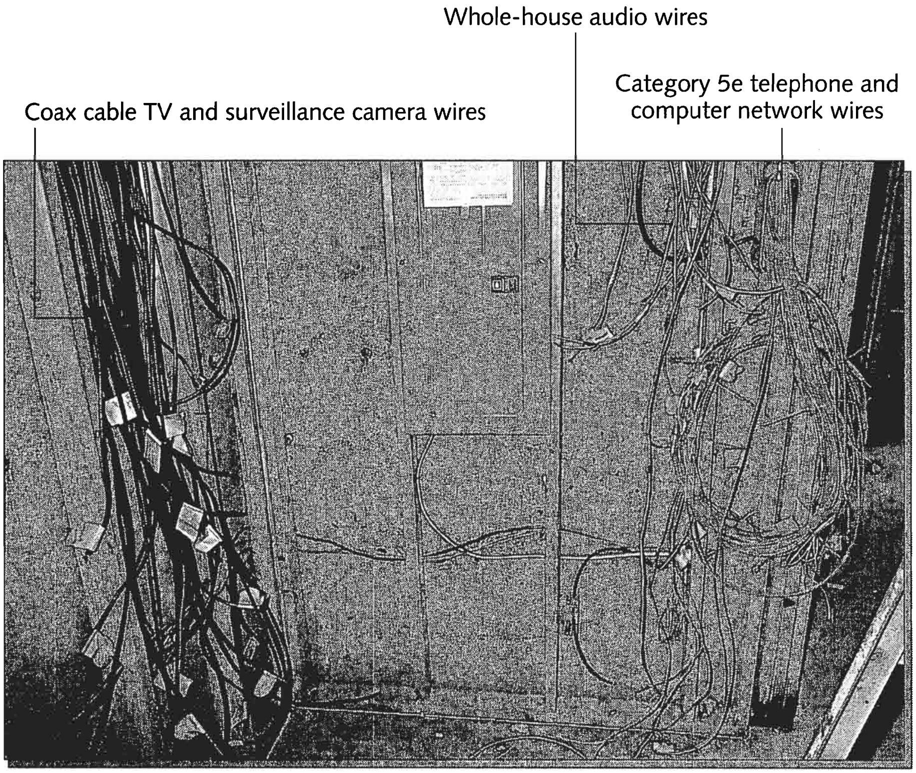


FIGURE 7-2: Drop cables to one spot for a Linux Toys Home Network server.

Wiring Tools

If you buy your cables, you can just plug them into each computer on one end, the hub or switch on the other, and go. However, if you are going to make your own cables, you need the following tools and components:

- **RJ-45 crimping tool** — This is used to connect the RJ-45 connectors to the Category 5e cable. Both the hub and the Ethernet cards are female connectors, so you can add male RJ-45 connectors to each end (unless you are adding a wall receptacle or a patch board).
- **Network Cable Tester** — Get a network cable tester that can test UTP cables.
- **Wire strippers** — You need a tool to strip the outside covering off of the cable.

Premade cables are appropriate for connecting only a few computers in the same room. Once you begin having longer runs around your house or office, you will want to make your own custom-length cables.

Step 2: Connect it all

If your computers are all in one place, simply connect a cable to an Ethernet card on each computer; then plug the other end into the hub or switch (depending on which you are using). Figure 7-3 illustrates what your Linux Toys Home Network might look like.

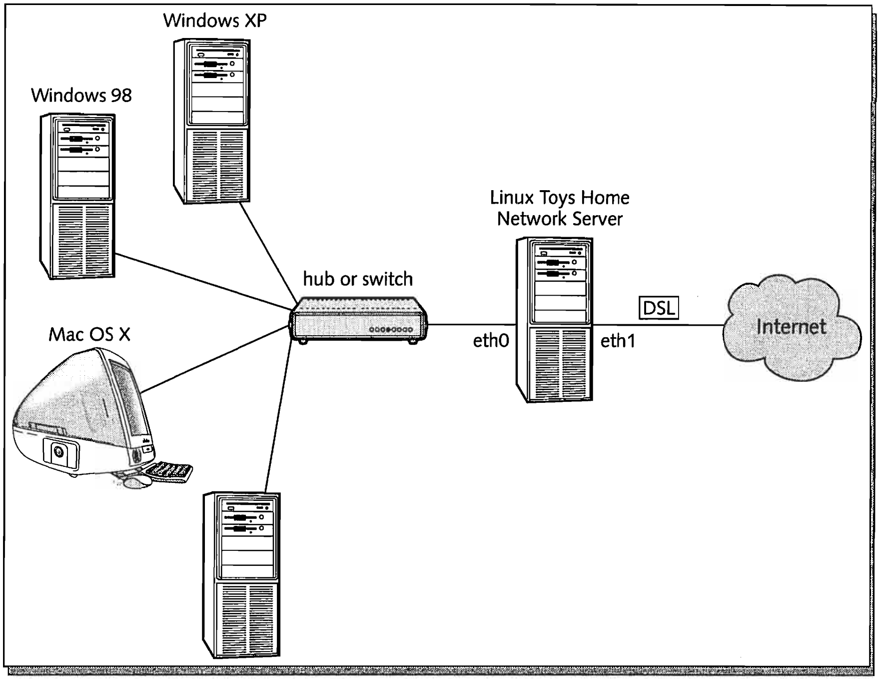


Figure 7-3: Connect each computer's Ethernet card to the LAN hub or switch.

This example shows multiple computers (Windows 98, Windows XP, Mac OS X, and Linux) all connected together into a hub or switch. The Linux Toys Home Network server computer has a connection to both the hub/switch and the equipment that provides the connection to the Internet. It also has a printer connected to it that can be shared by users on the LAN.

In the following sections, I describe how to configure your server to act as a firewall, router, and file and print server for your LAN. Then I describe how to add your Linux Toys computers to this network.

Configuring the Home Network Server

Your Linux Toys Home Network server can be the center of your home computer network. In our example, all data that travels between your home network and the Internet is routed and filtered through this server.

Because we're cheap, we don't want to have to buy another computer (or leave more than one computer on all the time). So, besides routing/firewall functions, our server will manage file and printer sharing among our home computers.

The Linux Toys Home Network server that we have you configure here is not a public server. What I mean is that you will not be serving Web pages, e-mail accounts, or FTP directories to people trying to access your server from the Internet. It is meant for you and your family to visit the Internet safely but to keep the bad people on the Internet from visiting you.

The features we describe in this section come with Red Hat Linux. Those features include:

- **Firewall (iptables)** — Out of the box, Red Hat Linux uses iptables as its firewall software. Besides deciding which packets do and don't get through, iptables lets you use features such as Network Address Translation (NAT) or IP Masquerading, which allow computers on your private network to communicate on the Internet.
- **Dynamic addresses (dhcp)** — We want the computers on our Linux Toys Home Network to connect to the Internet without much fuss. By setting up our server to be a DHCP server, the computers on our LAN (Windows, MAC, or other Linux systems) can just fire up and be on the Internet with minimal configuration.
- **File and Print Sharing (Samba)** — While there are several different software package for sharing printers and files on a network, the most popular is Windows file and print sharing, using the Server Message Block (SMB) protocol. The software that implements SMB in Red Hat Linux is called Samba.

Although in Chapters 11 and 12 we have you edit plain-text configuration files to get the servers going, we won't make you do much of that here. Both iptables and Samba have nice graphical interfaces for configuring firewalls and file/print sharing. The Security Level Configuration window offers a simple means of restricting access from the Internet, while trusting computers on your LAN. The Samba Server Configuration window lets you share directories and indicate the type of security to use.

Figure 7-4 shows what our Linux Toys Home Network server can do.

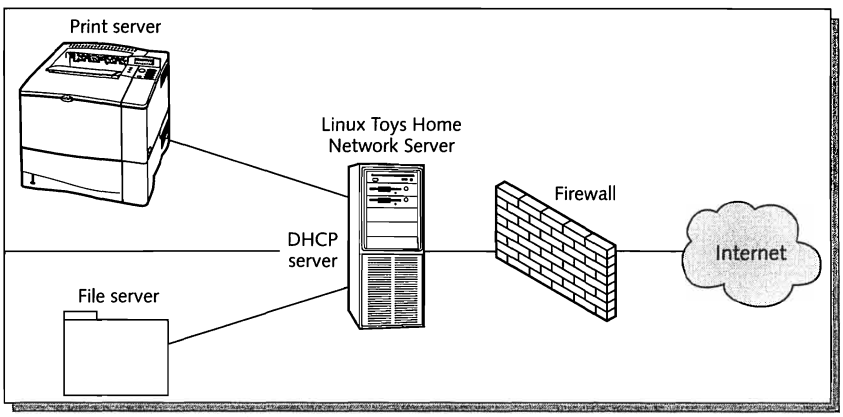


FIGURE 7-4: The server's firewall blocks intruders while happily sharing files and printers for your family.

Note

If you have a Mac OS X or earlier Mac computer on your network, refer to *Red Hat Linux Bible* for setting up a Mac to access Linux servers in a variety of ways. (I added a chapter on this topic to *Red Hat Linux 9 Bible* and later editions that includes how to set up Linux as an AppleTalk server using netatalk.)

Step 1: Install and configure Red Hat Linux

Start with a Server install type of Red Hat Linux and add GNOME or KDE desktop packages (see Appendix C) so that you have a GUI. If you want to do a minimal installation, you can create the configuration files by hand (no GUI). That can save you about 1GB of disk space. Important packages that you will need include the following:

- **redhat-config-samba** — Contains the Samba configuration window
- **samba-common** — Contains the configuration files and commands needed to use Samba file and printer sharing
- **samba** — Contains extra commands and documentation used with Samba
- **gnome-lokkit** — Includes the Lokkit (Configure Firewalling) window
- **dhcp** — Contains features to configure Red Hat Linux as a DHCP server

Before you start installing Red Hat Linux, you should install two Ethernet cards on the computer. This lets you begin to configure the following features during Red Hat Linux installation:

- **Network configuration** — Defines how Red Hat Linux gets the names and addresses it needs for your Ethernet cards to communicate with the Internet and your LAN
- **Firewall** — Sets the basic firewall rules that allow or prevent different kinds of services to pass between the Internet and your LAN through the Linux Toys Home Network server

Refer to the Create Network Interfaces (server and client) and Set up the Firewall steps for information on how to set up those two features (either during installation or after Red Hat Linux is installed).

Step 2: Create network interfaces (server)

We recommend two Ethernet cards on your Linux Toys Home Network server. One connects the server to the Internet and the other connects it to your home local area network (LAN). Red Hat makes it easy to set up those interfaces.

On the LAN side, I can suggest exact ways to set up the network information that your server and the client computers on your LAN need. On the Internet side, there is some variation, depending on how you are connecting to your Internet Service Provider (ISP). Here is the information you need to get started:

- **LAN interface** — We are going to have every computer on your LAN assigned its own IP address, using private IP addresses. So the interface to your LAN from the server will be assigned the address 10.0.0.1 with a netmask of 255.255.255.0. This address allows you to use IP addresses 10.0.0.2 through 10.0.0.254 for the other computers on your LAN.

Note

If you already have a home LAN in place, you may need to use IP addresses that work with the pool of addresses that are in place. If you are starting your LAN from scratch, the private addresses recommended here will work well.

- **Internet interface** — We are assuming that you are connecting your Linux Toys Home Network server to some high-speed DSL or cable modem connection to get to the Internet. If that is the case, you need to connect and configure that equipment using the directions from your ISP. (Otherwise, see the "A Lesson in . . . Dial-out Internet Connections" sidebar.)

  What you need to know from the ISP is: When you connect the Ethernet card on your server to the DSL or cable modem, will that equipment have DHCP running to assign an address to your Internet interface? If it does, you can simply configure your network interface to use DHCP. (If you are not sure, you can simply configure the interface using DHCP and see if it picks up an address automatically.)

  If the ISP's equipment does not provide you with an address using DHCP, you need to assign a permanent IP address to your Internet interface. The ISP might give you a permanent IP address to use. Or, more likely, the DSL modem or other equipment might come with an interface with an address assigned, often 192.168.0.1. If that is the case, you can assign an IP address of 192.168.0.2 to your interface.

During Red Hat installation

During Red Hat Linux installation, if you have your Ethernet cards installed, you can configure them to work with your LAN from the Network Configuration screen that appears. Figure 7-5 shows an example of that window.

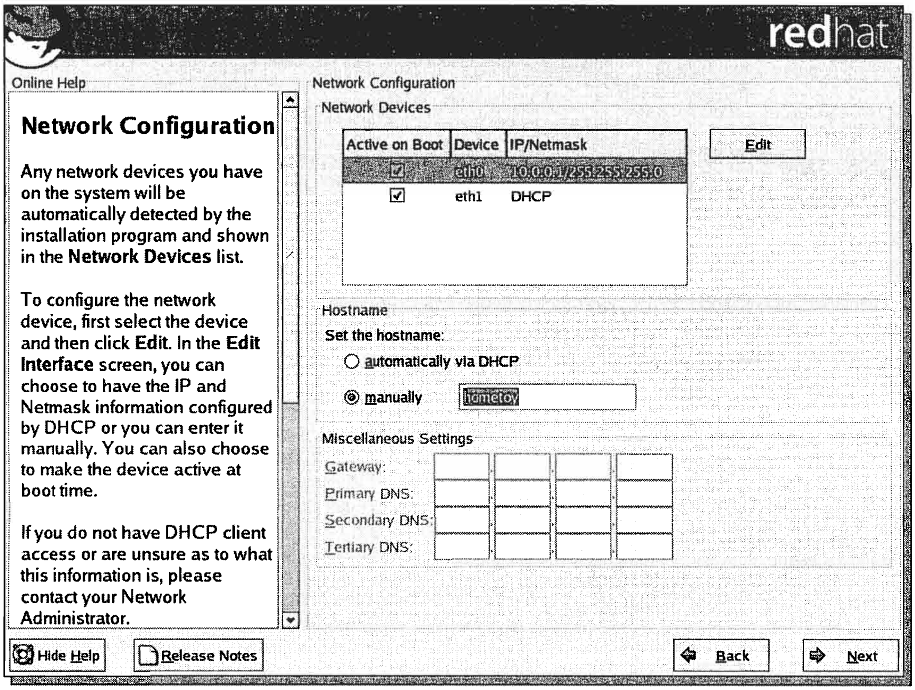


FIGURE 7-5: Configure your LAN and Internet connections during installation.

A Lesson in . . . Dial-out Internet Connections

Yes, you can use a dial-up modem as your Internet connection for the Linux *Toys* Home Network server. There are just a few drawbacks:

- **Slow as molasses** — You know how slow dial-up is from one computer. When a few people are using the connection at once, it will drag.
- **On-demand dialing** — Dial-up connections aren't made to be up all the time. So you need to configure your Red Hat Linux Home Network server to dial out when one of your computers tries to reach the Internet and to turn off after it has been idle for a while.

To adapt this project to use a dial-up Internet connection, skip the second Ethernet card. Then use the Network Configuration window to configure outgoing modem connections (click System Settings→Network from the Red Hat menu; then choose Add.) Later, when I describe how to set up routing, you will route to the ppp0 interface you just created (instead of eth1).

To configure demand-dialing to dial up your ISP each time someone on your LAN tries to get on the Internet, modify the `/etc/sysconfig/network-scripts/ifcfg-ppp0` file so that these lines appear at the beginning:

```
ONBOOT=no
DEMAND=yes
IDLETIMEOUT=900
RETRYTIMEOUT=30
```

This causes your modem to connect only on demand. After 15 minutes (900 seconds) of inactivity, the modem disconnects. If the modem disconnects when you want it connected, another dial-out attempt is made after 30 seconds.

Both Ethernet cards should appear on the Network Configuration screen. In our example, I'm using the first Ethernet interface (eth0) to connect to the LAN and the second (eth1) to go to the Internet. Here's how:

1. Click eth0; then select Edit.
2. Click to remove the check next to Configure Using DHCP.
3. Type the Ethernet address as **10.0.0.1** and netmask as **255.255.255.0**.
4. Click OK.

5. If your ISP is using DHCP to give you an IP address for your Internet interface (eth1), you don't have to do anything (it should already be set to use DHCP). Here are the other possibilities:
   - **Permanent IP address** — If the ISP gives you a permanent IP address (and netmask), you should enter that information, following the steps you did for the eth0 interface.
   - **Device has an IP address** — It might be that your DSL or cable-modem box is set to an address, often 192.168.0.1 with a 255.255.255.0 netmask. If that is the case, you should enter an address for the interface to that device that works on the same subnetwork (for example, 192.168.0.2 with a 255.255.255.0 netmask).
6. Click the Set the hostname: manually box and type a name for your computer. I'm assuming that you are not setting up a domain name for your LAN, so you can just give it a name you like. For example, you could call it hometoy or ltserver.
7. If you are not using DHCP to get your network address, you need to select the interface and type the following information:
   - **Gateway** — Type the IP address of the computer or router that provides the server's access to the Internet (over eth1, in the example). Again, this could be the IP address the ISP assigned to you or 192.168.0.1 (in the example of a DSL modem with a hard-coded address).

Note

The gateway address from the server is different from the gateway address of the clients on your LAN. The clients will all use 10.0.0.1 as their gateway, so that your server is identified as the router.

   - **Primary DNS** — The IP address of the primary DNS server (provided by your ISP)
   - **Secondary DNS** — The IP address of a secondary DNS server (also provided by the ISP), in case the first one is down

After Red Hat installation

If you didn't configure your server's network interfaces during Red Hat Linux installation (maybe you added the cards afterwards), you can do it from the Network Configuration window.

To open the Network Configuration window from the Red Hat menu, click System Settings → Network. Figure 7-6 shows an example of the Network Configuration window (before I installed the second Ethernet card and rebooted).

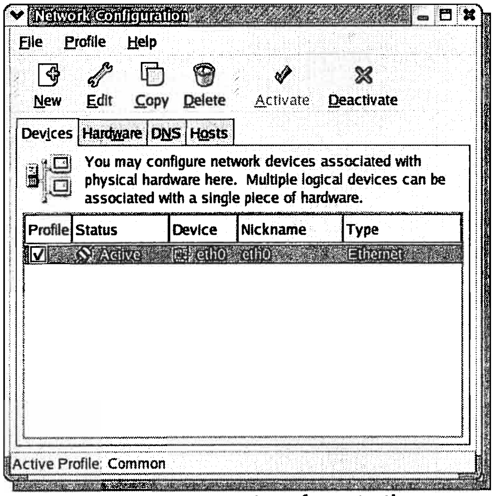


FIGURE 7-6: Manage your interfaces to the Internet and your LAN from the Network Configuration window.

As with the Network Configuration screen during installation, all installed Ethernet cards should appear in the window. Click the one you want to edit, and select Edit. Then add information about the interface as I described for setting up the interface during Red Hat Linux installation.

Step 3: Configure DHCP

Instead of going to each computer and typing the IP address, location of DNS servers, and other information that each computer needs to start up on your LAN, you can have that information distributed from your Linux Toys Home Network server. To do that, you need to configure the server to do DHCP.

You are going to have to edit a configuration file to do this. The file is `/etc/dhcpd.conf`. Then you have to start the `dhcpd` service. Here's how:

1. **Edit `/etc/dhcpd.conf`** — Go ahead and open this file in any text editor, as root user from a Terminal window. (Note that this file will not exist until you create it.) Then, following our example network, add these lines:

```
ddns-update-style none;
ignore client-updates;

subnet 10.0.0.0 netmask 255.255.255.0 {
    option routers              10.0.0.1;
    option subnet-mask        255.255.255.0;
    option domain-name-servers  123.45.67.1 123.45.67.2;
    option netbios-name-servers 10.0.0.1;
    range 10.0.0.2 10.0.0.200
```

```
        default-lease-time 21600;
        max-lease-time 43200;
}
```

You can use the configuration file shown here, with the exception that you *must* change the domain-name-servers IP addresses shown (123.45.67.1 and 123.45.67.2) to reflect the DNS server addresses from your ISP.

After that, here are a few other things to consider:

- **Lease time** — A lease in DHCP is the amount of time that a client gets to use an IP address before renewing it. The default lease time is set to 21600 seconds (six hours), but the client can request a lease time of up to 43200 seconds (12 hours). If the client doesn't renew the lease before the lease time expires, the DHCP server can reclaim it. On a large network, you might make these values larger (so the server doesn't get hit on too much). If you like, you could use 2592000 and get 30-day leases. In Red Hat's preconfigured dhcpcd, the value is 86400 (one day).
- **DNS updates** — If you set up a DNS server that manages the computers on your LAN, as addresses are assigned to the client computers on your network using DHCP, you can pass that information to your DNS server. This is a very cool, very powerful feature. It lets you address the computers on your LAN by name, even though their IP addresses may keep changing.

  Because we are not setting up a DNS server for this project, we turn off this feature (`ddns-update-style none`). If you decide to go it alone, however, and set up DNS for your LAN, set the `ddns-update-style` to `interim` to turn on this feature. (Type `man dhcpd.conf` and read the section entitled "The Interim DNS Update Scheme" for further information.)
- **IP address range** — Using the 255.255.255.0 netmask with a 10.0.0.0 IP number, we assign 10.0.0. as the network number. Excluding 0, 1 (our server), and 255, it leaves us with an address range of 2 through 254. We're going to use addresses 10.0.0.2 through 10.0.0.200 to assign to any computer on the LAN that asks for an address. Until you start adding your toaster oven and electric toothbrush to your network (which is still a few years off), 199 client IP addresses should be plenty for you.

Note

For the moment, I'm going to hold aside addresses 10.0.0.201 through 10.0.0.254. We are going to use some of those later to assign to specific toys. If you want to later set up the dial-in server described in Chapter 11, I have you use a different set of addresses (including 10.0.1.10 and 10.0.1.11) so the two projects can co-exist on the same computer.

- **The Home Network server** — In our example, the Linux Toys Home Network server acts as a router and as the WINS server for your Windows file and print sharing. So we assigned `option routers` and `option netbios-name-servers` to our server's address (10.0.0.1).

2. Edit the `/etc/sysconfig/dhcpd` file to add the name of the Ethernet interface as an argument to the `dhcpd` daemon. This is to make sure that only computers on your LAN are assigned addresses from DHCP. If the LAN interface is eth0, change the DHCPDARGS line to appear as follows:

```
DHCPDARGS=eth0
```

3. **Start DHCP** — You can start the dhcpd server and set it to start immediately. Here's how:

```
# service dhcpd start
# chkconfig dhcpd on
```

Step 4: Create network interfaces (clients)

When an Ethernet card is installed on a Windows, Mac, or Linux computer, the computer typically assumes that you are using DHCP to get the address information you need to start up automatically on the network. Since you just created a DHCP server, your clients should just be able to start automatically and grab this information:

- An IP address (taken from the range you defined)
- DNS server addresses (that you got from your ISP)
- Gateway (the address of your server)
- WINS server address (again, the address of your server)

Otherwise, I just suggest that you be sure to set a hostname for each computer on your LAN. Those hostnames will be the names you use to identify the shared printers and folders.

Step 5: Set up the firewall

The firewall you configure on your Linux Toys Home Network server is used to set a boundary between your private computers and potential intruders from the Internet. Using rules you set up, the firewall can decide what to do with packets coming in, going out, and being forwarded through your server.

There are a lot of features built into the iptables facility, which is the default firewall software for Red Hat Linux. You are primarily concerned with the following features:

- **Allow outgoing connections** — You want the firewall on your Linux Toys Home Network server/firewall/router to *trust* and allow communications from the computers on our LAN.
- **Restrict incoming connections** — You want to restrict communications that are initiated by computers outside of your LAN (that is, from the Internet), to the fewest possible services you need.
- **Do network address translation** — Because you are using private IP addresses on your LAN, for computers using those addresses to communicate on the Internet, their communications must appear to be coming from a real IP address. Because your server has a real outgoing address on its interface to the Internet, you can use Network Address Translation (NAT) in the firewall to make all packets from the LAN to the Internet look like they are coming from the server.

I struggled with how to suggest you set up your firewall initially. To get your initial setup, you can use two different GUIs that come in Red Hat Linux:

- **During installation** — The Firewall Configuration screen appears during installation of Red Hat Linux.
- **After Red Hat installation** — A similar screen appears by opening the Security Level Configuration window after Red Hat Linux is running. (Click System Settings ➔ Security Level from the Red Hat menu to display this window.)

Figure 7-7 shows the Firewall Configuration screen that appears during installation.

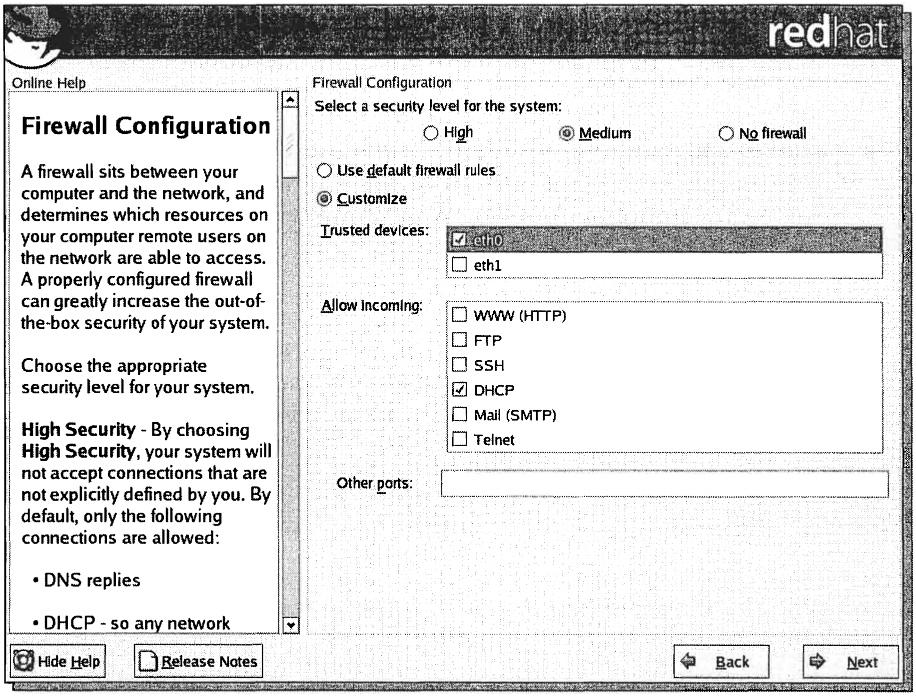


FIGURE 7-7: Set your firewall configuration when you install Red Hat Linux.

Unfortunately, none of the firewall configuration screens let you set up Network Address Translation (which we need for this project). So the approach I take here is to set an initial firewall configuration using one of the two GUI screens. Then edit the firewall configuration file (`/etc/sysconfig/iptables`) by hand to add the NAT stuff.

Caution: After you edit the iptables file manually, don't edit it again using any of the Firewall Configuration screens. If you do, it will erase the NAT firewall rules we have you build (along with everything else). Just use the GUI to give you a nice set of rules to start with.

Configure the firewall initially

Using the Firewall Configuration screen (either during installation or by starting the Security Level Configuration window with Red Hat Linux running), here are the settings I recommend you start with:

- **Customize** — Select Customize so you can set some custom firewall rules.
- **Trusted devices** — Assuming that eth0 represents the interface to your LAN, click eth0 in the Trusted devices box. This way, any requests coming from your LAN won't be blocked from the firewall. Besides allowing the computers on the LAN to browse the Web and get e-mail, it will allow them to use other services that you offer from the server (such as Samba file and print sharing).
- **Allow incoming** — By default, you don't want communications initiated outside of the LAN to be accepted. The one exception is DHCP. If your server gets the IP address for its interface to the Internet from an ISP's server, you should click the check box next to DHCP.

Add NAT to the firewall

You can't communicate directly to the Internet with private IP addresses, such as the 10.0.0.*X* addresses we recommend. For that reason, we have to set up our firewall to do what is called Network Address Translation (NAT). NAT translates the private addresses so that they appear to come from the one public address representing our connection to the Internet.

To do this, we have to add a few lines to the `/etc/sysconfig/iptables` file that contains our firewall rules.

Note: If you are using a dial-out modem as your connection to the Internet, instead of NAT you could use a feature called IP masquerading. The common belief is that IP masquerading works better on interfaces where the address is set dynamically, such as dial-up interfaces or those set by DHCP.

Here is an example of a line to add to the iptables file to make NAT work for your LAN. Assuming that your interface to the Internet is on eth1, and you are connected to a device (such as a DSL modem) that is at address 192.168.0.1 (and your address is 192.168.0.2), add this to the end of the file:

```
-A POSTROUTING -o eth0 -j SNAT --to-source 192.168.0.2
```

If your ISP gave you a permanent IP address to use for your Internet interface, you could use that instead of 192.168.0.2. For a dial-up Internet interface to the Internet, you could use an IP masquerading line instead. Here is an example:

```
-A POSTROUTING -o ppp0 -j MASQUERADE
```

The line just shown indicates that the IP address assigned to your dial-out interface (ppp0) is used to translate your LAN's network addresses. You can also use the `MASQUERADE` line if

your Ethernet interface to the Internet receives its IP address from your ISP's DHCP server. (In that case, you would substitute eth1 for ppp0.)

Poking holes in your firewall

Our Linux Toys Home Network server's firewall is made to keep traffic out of our network that did not originate on the inside. Because of that, some services that you try to use, such as active FTP and IRC chats (which can cause a remote server to initiate a connection to you), can fail.

Chuck's Tech Tip

When an FTP file request is initiated, the FTP server initiates a separate connection back to the client machine to facilitate the actual data transfer. This reversed connection from server to client is completely new and includes the traditional three-way TCP/IP handshake. From a firewall standpoint, this sort of connection would not be allowed back through.

There are two solutions to this. One is called FTP passive mode (which users can set from their own FTP clients); the other is the ip_conntrack_ftp module (which you can set up as an administrator on the server). The ip_conntrack_ftp module keeps track of ftp data requests and allows ftp data connections initiated from the server to connect through the stateful firewall to the client. A simple way to ensure that this module is loaded at boot time is to add a modprobe line to your rc.local file for ip_nat_ftp and ip_conntrack_ftp.

Start up iptables

With your firewall rules in place, you need to restart iptables to have the rules take effect. As root user from a Terminal window, type the following two commands to set iptables to start automatically and restart immediately, respectively:

```
# chkconfig iptables on
# service iptables restart
```

Step 6: Set up routing

In order for your Linux Toys Home Network server to route packets from your LAN to the Internet, you have to tell the server that it is all right to do that. First add the following line to your `/etc/sysctl.conf` file (as root user, using any text editor):

```
net.ipv4.ip_forward = 1
```

Then, if you are doing masquerading as well (as opposed to NAT), also add the following line to the same file:

```
net.ipv4.ip_dynaddr = 1
```

Those features will be turned on the next time you boot your computer. To turn them on immediately (so you don't have to reboot), you could type the following from a Terminal window:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
# echo 1 > /proc/sys/net/ipv4/ip_dynaddr
```

Step 7: Set up Samba file and print sharing

Because one of our requirements for this project was to not have to keep more than one computer on all the time, we're going to have our Linux Toys Home Network server act as a file and print server as well as a firewall/router. There are several advantages to a central file/print server:

- You can spend your money on one nice printer, instead of having three different junky ones around the house.
- You can have one massive hard disk, containing all the family's photos, music, and videos. All sharing and backups are done from one place.

The Samba Server Configuration window provides a nice interface for configuring your computer as a Samba file and print server. Log in to the Linux Toys Home Network server you are configuring and do the following:

1. Add Linux Users — Add a user account for each person who has a computer on your LAN. To simplify setup, you can have the user name and password match the ones they use on their own computers. (See Chapter 12 for a description of adding user accounts.)
2. Add a Linux Printer — From the Red Hat menu, click System Settings ➔ Printing. The Printer configuration window appears.
3. Click New. The Add a new print queue appears.
4. Follow the instructions for adding a printer. (For our purposes, you can set it up as either a networked UNIX or Windows printer. Either should work.)
5. Select the printer from the list; then select Action ➔ Sharing.

Note

If your printer is fairly new, it may not appear on the list. Often, you can simply select PostScript Printer, if your printer supports PostScript. Or you can configure a Raw Print Queue and have each client use its own print driver (the printer probably has Windows drivers for the printer that your clients can use). Otherwise, check out `linuxprinting.org` to get help.

6. From the Sharing properties window that appears, click the box next to This queue is available to other computers. Then click OK.
7. Click Apply. This printer will automatically be available through Samba, once Samba is configured.
8. Configure the Samba Server — Open the Samba Server Configuration window from the Red Hat menu by clicking System Settings ➔ Server Settings ➔ Samba Server. Enter the root password, when prompted, and the window appears.
9. Click Preferences ➔ Server Settings. The Server Settings window appears.
10. Set the following information on the Basic and Security tabs:
    - **Workgroup** — Create a workgroup name that will be used by all the computers on your LAN.
    - **Description** — Type a few words describing the server, such as Home Network Server.

On the Security tab, I suggest that you leave Authentication Mode (user), Authentication Server (blank), Encrypt Passwords (Yes), and Guest Account (No guest account) as they are. Click OK to continue.

11. Click Preferences ➔ Samba Users. The Samba Users screen appears.
12. Click Add User. The Create New Samba User window appears.
13. I suggest that you create a Samba user account name and password that match the Linux user name and password you created for each user in Step 1. Here is what you add:
    - **Unix Username** — The name of the user account you created in Linux. There should probably be one representing the user of each client computer.
    - **Windows Username** — I recommended that the Unix and Windows names be the same, for simplicity sake.
    - **Samba Password** — Again, I recommended that you use the same password for the user as you did for the associated Linux user name.

    Click OK to continue. Then repeat this step for each user.
14. Create Samba Shares — Click the Add button. The Create Samba Share window appears.

    In this set of steps, you want to identify each directory you want to share and who can access it. One strategy is to share the home directory of each user so that only he or she can access it. You may also want to create a central repository, from which family images, music files, documents, and videos can be shared.
15. From the Basic and Access tabs, add the following information:
    - **Directory** — The name of the directory you want to share (such as `/home/chris` or `/var/family`). The directory must exist. (For example, as root user type `mkdir /var/family` from a Terminal window.)
    - **Description** — A short description of the directory
    - **Basic Permissions** — Click Read-only or Read/Write. If you select Read/Write, Linux permission must also allow writing. (To check and open Linux permission, use the `chmod` command on the files and directories you want to share.)
    - **Access** — On the Access tab, select either Only allow access to specific users or Allow access to everyone. If you are sharing a `/home` directory (such as `/home/chris`), you probably want to allow access to only the user who owns that home directory. For a shared directory, you might want to add everyone in your family or simply click Allow access to everyone.
16. Click File ➔ Quit to quit the Samba Configuration window.
17. Restart the Samba service — As root user from a Terminal window, type the following:

```
# chkconfig smb on
# service smb start
```

At this point, the shared printers and directories that you added should be available to clients on your network. Open the Network Neighborhood or Network Places window on your Windows computer to see the directories (folders) and printers that are available from your Linux Toys Home Network server.

Making Your Network Toy-Ready

So far, you have created a network where you can safely (and pretty easily) get to the Internet and share files and printers among the computers on your home network. Any computer that you connect to your LAN that is set to use DHCP (which most are) will automatically gain instant access to the Internet. To be able to have your server and Toys talk to each other, however, you have to do a bit more work.

As it stands, when a computer starts up on your LAN, it is assigned an arbitrary IP address, with no TCP/IP name attached to that address. To be able to reach our Linux Toys computers (some of which may be running without a keyboard and monitor), however, we need to know the IP address or, even better, know a name that can be used to reach that address. There are a few ways to go about this. Here is how we're going to have you do it:

- **Fixed IP addresses** — When DHCP assigns an IP address to a computer on your LAN (actually, to its Ethernet card on your LAN) you can tell it to assign a particular IP address. You do this by associating a MAC address (which uniquely identifies the Ethernet card) to an IP address.
- **Names in `/etc/hosts` file** — Assuming that you will use your Linux Toys Home Network server to communicate with the other Linux Toys computers in your house, you can add the host name and IP address to your server's `/etc/hosts` file for each of those computers. Then you can distribute that file to the Linux Toys computers so they can all reach each other.

Now, to all networking gurus who begin hopping up and down and throwing this book against the wall at the mere mention of using the `/etc/hosts` file, let me explain why I did it this way. We're talking about a home network here, where I'm assuming that the configuration will look something like the one shown in Figure 7-8.

In this example, most of the computers on the LAN are just looking to surf the Web and share files and printers. They need only to reach other computers for file and printer sharing (which they can do fine as described in the Samba File and Print Sharing step). The computers that need to be contacted using TCP/IP, in this model, are all Linux computers. To save everyone the trouble of setting up DNS for a home LAN on this project, you can do this instead:

- **DHCP** — Assign particular IP addresses to the Linux Toys and Linux Home Network server computers on the network.
- **`/etc/hosts`** — Create an `/etc/hosts` file and add the names and IP addresses for all Linux computers on the LAN.

Note

You certainly can set up dynamic DHCP and DNS on your LAN to do what we are doing here more elegantly. It will scale up better when you have hundreds of Linux Toys on your LAN. (Go for it!) I devote more than a dozen pages to a similar DNS configuration in *Red Hat Linux Bible*, but I think it's overkill for what we are doing here.

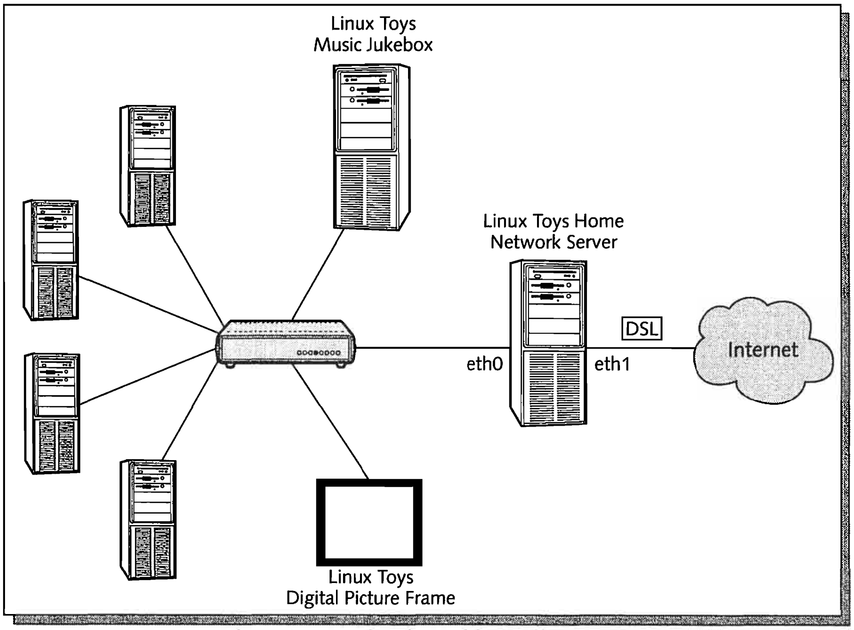


FIGURE 7-8: Here's your home network, after adding a few toys.

Here is what I did to assign permanent IP addresses to my Linux Toys and server computers and to then distribute those addresses to each of those computers:

1. **Determine MAC addresses** — For each Linux computer on your network, you want to determine the MAC address of the Ethernet card connecting to your LAN. To do this, boot the computer (it should temporarily get an address from your DHCP server). Then log in and type the following from a Terminal window:

```
# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:0B:6C:02:EC:94
          inet addr:10.0.0.100  Bcast:10.0.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:9126 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3351 errors:5 dropped:0 overruns:0 carrier:5
          collisions:4780 txqueuelen:100
          RX bytes:9561679 (9.1 Mb)  TX bytes:235862 (230.3 Kb)
          Interrupt:11 Base address:0xdc00
```

   You should see an eth interface for each Ethernet card (eth0, eth1, and so on) and a loopback interface (`lo`, not shown here). What you want to write down is the Hwaddr (in this example I highlighted the address: `00:0B:6C:02:EC:94`) for the Ethernet interface that is connected to your LAN.

2. **Edit /etc/dhcpd.conf**— Open `dhcpd.conf` with any text editor, as root user. Then add these lines to that file (substituting the names of your Linux computers and MAC addresses for the ones I show), making sure to not miss a semicolon (;) or brace ({}):

```
host ltserver {
     hardware ethernet 00:0B:6C:02:EC:94;
     fixed-address 10.0.0.1;
}
host ltmusic {
     hardware ethernet 00:50:DB:7A:97:7C;
     fixed-address 10.0.0.201;
}
host ltpicframe {
     hardware ethernet 00:D0:E7:89:A5:33;
     fixed-address 10.0.0.202;
}
```

Notice that I started with addresses 201 and 202, because they are after the address range assigned to be automatically distributed by DHCP (2-200). I'm calling the Linux Toys Home Network server `ltserver`. I then name my music jukebox and digital picture frame `ltmusic` and `ltpicframe`, respectively. (You can type `uname -n` on each Linux computer to see the name you give to it.)

Note

If you change any Ethernet cards in the future, you will need to update the `dhcpd.conf` file to show the new card's MAC address.

3. **Edit /etc/hosts**— Open the `/etc/hosts` file (as root user, with a text editor) on your Linux Toys Home Network server and add the following lines:

```
10.0.0.1 ltserver
10.0.0.201 ltmusic
10.0.0.202 ltpicframe
```

Of course, you want to use the host names and IP addresses to match the computers you have configured. Note that the IP addresses here must match the ones you are assigning in your `dhcpd.conf` file.

4. **Distribute /etc/hosts**— Assuming that sshd is running on each Linux computer on your network (and that you have the root password to each machine), you can type the following command line to copy your updated `/etc/hosts` file to each of those computers from the server.

```
# scp /etc/hosts root@ltmusic:/etc/hosts
root@ltmusic's password: *******
hosts     100%|***********| 165   00:00
```

Repeat this command to copy the `/etc/hosts` file to each of your Linux Toys computers, substituting each computer's name (such as `ltpicframe`).

At this point, you should be able to communicate among all the Linux computers on your LAN, identifying each computer by its hostname.

Summary

You can leave your Linux Toys Home Network server up and running all the time (or turn if off when you don't expect anyone to use the network for an extended period). While it's running, it will act as a firewall, router, and boot server (DHCP), so the client computers on your LAN can immediately connect to the Internet.

Your Linux Toys Home Network server can be the focal point for the rest of your home network. It can become a Windows file/print server (using Samba). It can also be a place where you can log in and access any Linux Toys on your network (such as the music jukebox or digital picture frame) and change songs, playlists, or images on those computers.

chapter 8

Making a Home Broadcast Center

in this chapter

☑ Setting up hardware and software

☑ Opening your firewall to stream video

☑ Configuring streaming video

☑ Broadcasting video and audio locally or on the Internet

There's no reason why any video that comes into your Red Hat Linux computer has to stay there. Using tools such as ffserver and ffmpeg, other computers can connect to your Red Hat Linux server and view video that you stream out on the network. That network can be:

- Your local LAN (where you can pump out video to the LAN almost as fast as you can process it) or
- The Internet (where you can reduce the quality a bit, but still get decent performance across a DSL, cable modem, or better connection)

Think of the possibilities! You could:

- Set up a surveillance camera at home and keep an eye on your house from your computer at work.
- Let Grandma in Florida watch little Timmy's birthday party live from Washington.
- Broadcast a television program from your Red Hat server to your laptop as you sit in the garden.
- Start your own 24-hour-a-day, all-Chris, all-the-time Internet broadcast outlet from your basement.

Well, that last one will work if you don't have more than one or two people watching at a time. And if they don't mind that the video locks up every once in a while. And if your name is Chris. Remember, video in Linux is still in its early stages. But hey, you can get in on the ground floor!

This chapter steps you through setting up streaming audio and video on a Red Hat Linux server using tools from the ffmpeg package. It covers ways you can get video input (television, video cameras, and Webcams). Then it describes how clients on your LAN or the Internet can play the video that you are streaming out on the network on their computers.

Who knows? This project could mark the start of your Linux World News network (see Figure 8-1).

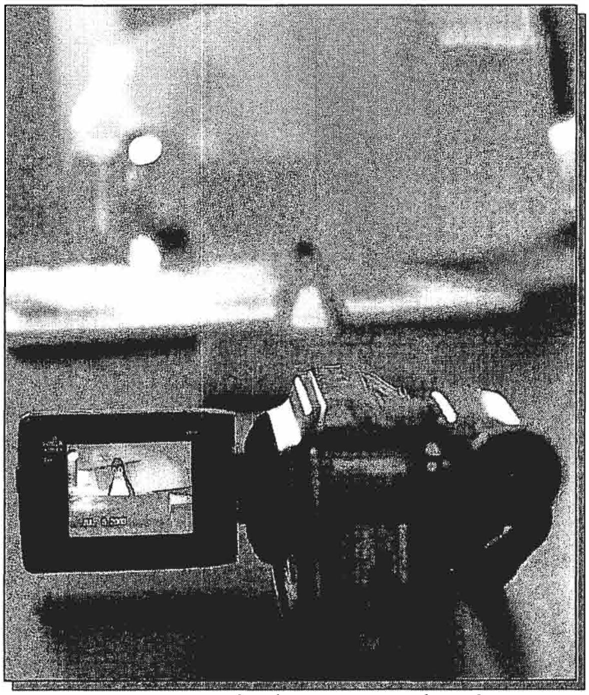


FIGURE 8-1: Broadcast video from your own first-class studios to very few people.

What's in the Home Broadcast Center?

The hardware and software you use for this project are very similar to those used in the Home Video Archive project in Chapter 4. The biggest differences are that, in Chapter 4, you copy video input to hard disk and burn it to CD or DVD, while in this project you take video input and stream it out to another computer over a network.

The ffmpeg project contains tools for both compressing the streaming video and acting as a server to handle client requests for the streaming video/audio. You can use xawtv to adjust the video. Then, depending on the type of computer you are playing the streaming video on, you can use one of a number of different players to play back the video stream.

You want a PC with a fast processor, a network interface card, and a video card and TV card that are up for the job. Beyond that, you need to decide if you are streaming television input from your antenna, cable TV, or dish to another computer in the house or connecting a video camera of some sort to capture the video you want to stream.

Step 1: Gather the hardware

For this project, you need to consider the hardware and software you need for both the server (which is streaming the video) and the client (which is playing the video).

The server has a camera, antenna, or cable TV connection to your TV capture card. The server has to be able to grab and stream the audio/video, but not necessarily view it. The client computer has to have a connection to the same network (whether it is the local LAN or the Internet) and have a player installed that can play streaming media in the format being used.

Review the issues described in Chapters 4 and 5 related to choosing a PC that you can use as the server for this project with video in Linux. Here are just a couple of additional notes relating to the computer resources you need for this project.

The personal computer

The main difference between the computing resources that you need for this project and those required for the video projects described in Chapters 4 and 5 has to do with disk space. While you are processing a lot of video in this project, you are not storing it permanently to hard disk. Therefore, you don't have to allow the approximately 2GB of disk space for each hour of video you record.

For good results, we recommend a Pentium 4, 1GHz processor and at least 1GB of DDR RAM. The project worked pretty well on my Pentium III, 650 MHz processor with 384MB of RAM. Shrinking the frame size and rate, we even got it to work on an AMD 400Mhz, with 64MB RAM (which is about the minimum needed to support Red Hat Linux).

Video card

You could use the same video card you used for projects in Chapter 4 or 5.

TV capture card

As with the projects in Chapters 4 and 5, you can use any television capture cards that are supported by video4linux drivers. You don't need a TV capture card if you are using a USB Webcam to get video to stream.

Sound card

The `ffmpeg` command expects there to be a sound card on the system you are streaming video from — even if you don't care to listen to the video on the server. So we recommend that you install and configure a supported sound card (see Chapter 3 for information on sound cards).

You can get around not having a sound card by using the NoAudio feature when you configure the `ffserver.conf` file (described later). Even running `ffmpeg --an` (to turn off audio) doesn't seem to work if you have no sound card. Also, you probably need to patch the Audio Out from the TV card to the Line In on the sound card using a short patch cable.

Video cameras

There are lots of different types of cameras you can use to produce the video you stream in this project. Some of your choices depend on the type of input jacks on your TV capture card. Here are some of the cameras you might consider using:

- **Camcorder** — Most video cameras for the home market have an A/V out jack that you can use to connect a cable between the camera and the TV card. If you don't have a cable that lets you directly plug the camcorder into your TV card, you can always plug it into a VCR and play video through there using a regular coax cable.

Note

We found that if you leave some camcorders on without actually recording, they shut themselves off after a while. To use the camcorder to record continuously (which they aren't really made to do), you need to find a way to disable the shut-off feature or possibly run some kind of continuous loop tape.

- **Webcam** — The Video4Linux drivers used in Linux support a limited number of Webcams. We've had luck with the Logitech 3000 and 4000 pro, as well as other models with similar chipsets. The problems we encountered were that images are not scalable and that the Webcam can interfere with your sound card. But otherwise, it works pretty well. For more information, you may want to check the video4linux documentation that comes with the Linux source code (kernel-sources packages, in the `/usr/src/linux/Documentation` subdirectories).
- **Security cameras** — Any security camera that can plug into a coax or RCA cable can be integrated into this project. To pick up inexpensive security cameras (under $100), you can try Web sites such as `smarthomes.com`. Some cameras use BNC connectors that connect to RCA output. If you are using it for a security camera, you can often go distances of up to 300 feet with these cameras.

Chuck's Tech Tip

Modern video cameras use a CCD (Charge Coupled Device) to translate images to electrical impulses. This is the same device the Hubble Space Telescope uses to record video. In essence, a CCD is a plate of silicon that reacts electrically to photons of light that hit it. The electrical signals are then translated to video by a signal processor on board the camera. Consumer-level video cameras as well as Webcams all create their video with CCDs. In order to keep Webcams affordable, their signal processor is of much lower quality when compared with a modern camcorder.

Speakers

Any PC speakers will do for this project. You can plug speakers directly into the TV capture card (when you are watching TV input) or to your computer's sound card (when you are playing back video). If you have patched your TV card into your sound card, you can just leave the speakers plugged into your sound card.

Ethernet card

The Ethernet card is needed to connect to the network. No dial-up modem will be fast enough to handle the streaming video/audio. You can broadcast television to other computers on your LAN if you don't have a DSL, cable modem, or better connection to the Internet.

Client computer

Because the streaming video is actually played on a computer besides the server, the client that plays the video must have a player that supports the file format, audio, and video codecs that ffmpeg is streaming. In Linux, you can use mplayer to play the streaming video. On a Windows computer, you can use Windows Media Player.

Step 2: Install Red Hat Linux

You can start with a Personal Desktop install of Red Hat Linux to do this project. You need to be sure to configure your Ethernet card to communicate over your network (either at installation time or later using the Network Configuration window described in Chapter 7).

Step 3: Install Linux Toys software

To make it easy to install the Linux Toys RPM packages you need for this project, they are all gathered in the ch08-HomeBroadcast directory on your Linux Toys CD. To install them, insert the Linux Toys CD and run the install.me script as follows:

```
# mount /mnt/cdrom
# cd /mnt/cdrom/ch08-HomeBroadcast
# ./install.me
```

You should be ready to start configuring your Home Broadcast Center.

Step 4: Install video card

If you already have a supported video card in your computer, skip to the next step. If you are adding a new video card, here is how you should go about adding that card:

1. Before installing the new video card, boot Red Hat Linux and log in.
2. Open a Terminal window and become the root user (type **su -** and enter your password when prompted).
3. Edit the `/etc/inittab` file to change the `initdefault` to 3. (This will keep you from having scrambled video if your video card is not automatically detected when you reboot.) You will probably change from 5 to 3 so the line appears as follows:

   ```
   id:3:initdefault:
   ```

4. Shut down your computer, unplug the power cord, and physically install the new video card (per manufacturer's instructions). Reattach the power cord and connect your monitor cable to the new card.
5. Boot your computer. As Red Hat Linux boots up, it should detect your new video card and ask if you want to configure it. Say that you do want to configure it.
6. When you see the text-based login, log in as the root user.
7. Type the following command to configure your video card:

   ```
   # redhat-config-xfree86
   ```

   If your video card was detected, you should see the Display Settings window.

8. You can adjust the following settings from the Display Settings window:
   - **Resolution** — Higher resolutions look better (and let you fit more on a screen), while lower resolutions can perform better.
   - **Color Depth** — Likewise, higher color depths allow more colors, but can slow performance (which can be an important issue when displaying video).
   - **Monitor Type** — The window tries to probe your monitor. If it can't detect the monitor type, you should go to the Advanced tab and configure it yourself. You need to refer to your monitor's documentation for the vertical and horizontal sync rates you need to use.
   - **Video Card Type** — Like the monitor, the Display Settings window tries to detect your video card. If it is not properly detected, go to the Advanced tab and select a video card (and its driver) from the list presented.

   When you are done, save the changes and close the window.

9. To check that your video card is working, log out as root, log in as a regular user, and type the following:

```
# startx
```

Refer to Chapter 4 for information on getting drivers for NVidia cards and for ways of checking that your card is working properly.

Step 5: Install a TV capture card and video equipment

Go through the following steps to install your TV capture card and other video equipment:

1. Shut down your computer and physically install the TV capture card (as described by the manufacturer).
2. When you reboot, kudzu should detect that the card was added and ask you if you want to configure it. Say that you do.
3. Connect your video camera to your TV capture card.

Step 6: Check your setup

At this point, you want to check that your setup is working properly. Cross your fingers and do the following:

1. Run the `xawtv` command. The procedure for starting and running `xawtv` is presented in Chapter 5. You should be able to immediately see the output from your video camera, security camera, or Webcam.

Note

When we connected the Webcam and started xawtv, it just displayed images from the Webcam. From the xawtv Options window, a Webcam entry appeared as a Video Source selection. Using the Webcam in this way, we were able to bypass using a TV card at all.

2. Adjust the color, brightness, and other values to suit your own eye.
3. When the video appears as you would like it, save the changes (leaving the channel where you were when you watched the video) and shut down xawtv.

Your setup should now be complete. You should now be ready to start setting up the streaming. If anything isn't working, refer to the "Troubleshooting Video" section in Chapter 5.

Setting up the Broadcast Server

There are two basic elements to serving and streaming video in this project. The first is the `ffserver` command, which listens for requests for streaming video from clients, and the other is the `ffmpeg` command, which grabs and compresses the video from the video devices on your computer and forwards them on to the client.

The diagram in Figure 8-2 shows the relationship between the ffmpeg process, the ffserver, and the mplayer (or other video player) client.

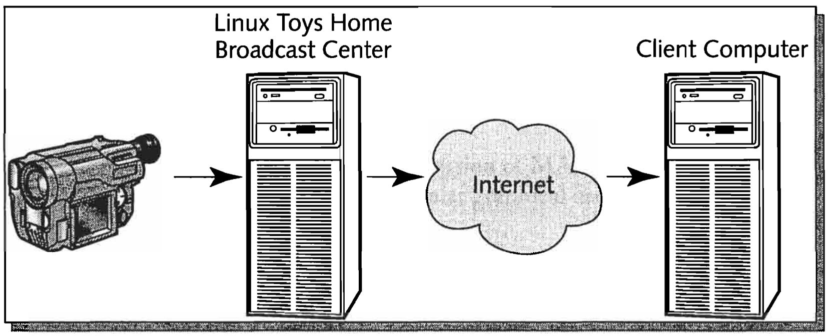


FIGURE 8-2: Stream video from your video source using ffmpeg, ffserver, and a video player.

Use the following procedure to set up your Linux Toys Broadcast server:

1. Configure your connection to the Internet or the LAN (depending on where you want to broadcast your streaming audio/video).

   Note: For the client computer to be able to reach your server from the Internet, you need either a fully qualified domain name (FQDN) for the server or an IP address. If you don't have an FQDN or a permanent IP address, you can give your clients your temporary IP address as the one they can use to reach your server. As root user, type **`ifconfig`** to see the IP address associated with your Ethernet interface (probably eth0).

2. If you are using a firewall (which you should if you are connected to the Internet), you need to punch a hole in it to allow clients to come into your computer on a particular port. The ffserver tool uses port 8090 by default. For an iptables firewall (which is the

Red Hat Linux default), you could add the following line to your /etc/sysconfig/iptables file:

```
-A RH-Lokkit-0-50-INPUT -p tcp -m tcp --dport 8090 --syn -j ACCEPT
```

3. Restart your firewall by typing:

```
# /etc/init.d/iptables restart
```

4. Next, create the /etc/ffserver.conf file. There is a sample ffserver.conf file in the /usr/share/doc/ffmpeg* directory. Copy it to /etc as root user from a Terminal window, as follows:

```
# cp /usr/share/doc/ffmpeg*/ffserver.conf /etc/
```

5. Now you can review and change the settings as needed. Because /etc/ffserver.conf contains settings that are forwarded to the ffmpeg command, you can modify these settings to change how ffmpeg performs.

   Here is an example of the /etc/ffserver.conf file, divided up into sections. (The file is loaded with comments, which I removed here for the sake of clarity.) The following settings are mostly default values, with notes where we changed values:

```
Port 8090
BindAddress 0.0.0.0
MaxClients 1000
MaxBandwidth 11000
CustomLog -
NoDaemon
```

   The Port option causes ffserver to listen for client requests from port 8090. BindAddress 0.0.0.0 causes ffserver to listen on any interface. (You can change that to an IP address of one of your Ethernet interfaces, if you have more than one.) MaxClients limits the maximum number of clients that can connect to the ffserver (no reason to limit this, since bandwidth is a better way to limit what resources clients can draw).

   The MaxBandwidth 11000 limits the amount of total bandwidth the computer can use for streaming video to about 11Mbps. We found that increasing MaxBandwidth (it was 1000 Kbps by default) can dramatically improve results (of course, it also consumes lots of computer resources). CustomLog - causes all error messages to be sent to standard out (you could replace the dash with a filename to log messages to a file). NoDaemon tells the server not to be run as a daemon.

   The next section describes the feed parameters:

```
<Feed feed1.ffm>
File /tmp/feed1.ffm
FileMaxSize 2000K
ACL allow 10.0.0.1 10.0.0.255
</Feed>
```

   The feed parameters just show details related to this particular feed. The feed is stored temporarily in the /tmp/feed1.ffm file, as it is being streamed. The FileMaxSize defines the maximum size of the file as 2000K (or about 2MB). (We increased the FileMaxSize from 200K.) The ACL line should be changed from the default 127.0.0.1

to whatever IP addresses you want to allow to see your streaming video. (We changed it to `ACL allow 10.0.0.1 10.0.0.255` to allow anyone on our local LAN to attach to the stream. We'll change it later when we are ready to go out on the Internet.)

The next set of options set how the stream is generated to handle the feed:

```
<Stream test1.avi>
Feed feed1.ffm
Format avi
AudioBitRate 32
AudioChannels 1
AudioSampleRate 44100
VideoBitRate 128
VideoFrameRate 30
VideoSize 512x384
VideoGopSize 12
</Stream>
```

The parameters in this snippet are among the most important for determining the performance and quality of your streaming server. These options set how each stream is generated to handle the streaming. The stream (which we renamed to `test1.avi`) feeds data into the `feed1.ffm` feed. The file format we changed to `Format avi`. Other settings that you will probably want to tweak to get your streaming audio/video just right are:

- **AudioBitrate** — This sets the bitrate of the audio stream to 32. To change this value, you should know what bitrates the codec you are using supports.
- **AudioChannels** — Choose 1 for mono sound or 2 for stereo.
- **Audio SampleRate** — Sets the sample rate for audio. We increased the value to 128 while streaming video on our LAN.
- **VideoBitRate** — Sets the video sample rate.
- **VideoFrameRate** — By default, this is set to 3 frames per second. We increased it to 30. You can reduce it if you like. However, a setting below 15 FPS causes the picture to appear jerky.
- **VideoSize** — Sets the size of each captured image. We set ours to 512 (pixels wide) by 384 (pixels high). Smaller images will perform better, but will be harder to see or won't scale. (The default is 160 × 128.)
- **VideoGopSize** — Sets VideoGopSize to 12, telling ffmpeg to do an intra frame every 12 frames. The intra frame consists of the entire image, not just the changes from the previous intra frame.

Note

If you run your server without a sound card, you can add the NoAudio option on a line by itself with the other Stream options just described to disable audio.

You should read the comments in the `/etc/ffserver.conf` file. You will certainly want to go back and make changes at some point to improve your streaming audio and video.

6. Turn on your video camera.

7. Run the xawtv command and select the output that you want to stream. There may be different selections you can make including Television, Webcam, or other types of devices. (I plugged my camcorder directly into the Video In on my TV card and selected Composite1 in xawtv.) Also, just double-check the color and brightness of the picture, and close the window.

8. To start the streaming server, type the following from a Terminal window on the server:

```
# ffserver
ffserver started.
```

9. Next start the fmpeg command to begin the streaming. Open another Terminal window and type the following:

```
# ffmpeg -ac 1 -vd /dev/video0 http://localhost:8090/feed1.ffm
```

   This tells ffmpeg to open one audio channel (-ac 1) and look for video to come from the /dev/video0 device (the default video driver). The http line tells ffmpeg to connect to port 8090 on the same computer. This stream is identified by the name feed1.fml. You should start seeing the streaming begin immediately (indicated by moving numbers under the command).

   The time should be right for client computers to try to reach your streaming audio/video.

10. To test the streaming audio/video from the local system, you could start playing the stream with mplayer. Open another Terminal window and here's an example of what you would type:

```
# mplayer -framedrop -cache 1024 http://localhost:8090/test1.avi
```

   The mplayer window should open and begin displaying the streaming video.

11. If you were able to play streaming video on the same computer, you can now make the streaming content more widely available. Edit the /etc/ffserver.conf file and change access restrictions so the server can be used outside of the localhost. For example, to make your streaming video available to all networks except 1.0.0.0, change the ACL allow line as follows:

```
ACL deny 1.0.0.0 1.255.255.255
```

12. On the client side, you can play the video from another computer (on the LAN or the Internet) by adding a host name or an IP address to the mplayer command line. For example:

```
# mplayer -framedrop -cache 1024 http://10.0.0.1:8090/test1.avi
```

   This line would get video from the machine at IP address 10.0.0.1 (presumably, the server on your LAN). For a public server, you would type a fully qualified domain name or the public IP address of the server. Then you can watch your security camera at home from your work.

Figure 8-3 illustrates video being streamed from one Linux system to several other Linux systems on a LAN. Because the clients are streaming audio and video from the server at different rates (from a 2MB buffer), the video that each sees may be in a different place.

FIGURE 8-3: Broadcast penguins over your LAN or the Internet with ffmpeg.

When you are done with the streaming video and audio, you can simply type **Ctrl+C** to stop the ffserver. Then do the same thing to stop the `ffmpeg` command.

If you want video to be constantly streaming, you can set ffserver to run as a daemon process. Once you type the `ffserver` command, it begins waiting for requests.

Tips for Using Your Home Broadcast Center

There are opportunities to improve your streaming video and audio from both the server and client side. Here are a few tips:

Improving the streaming video server (ffserver/ffmpeg)

We figured out that, in general, if you add up the audio and video bitrates and double them, you'll know approximately how much bandwidth you are going to consume from your server. For example, with video set to 128 kbps and audio set to 32 kbps (which together is 160 kbps), we found that one stream pretty much filled up the 256 kbps upload speed Chuck has on his cable modem service.

If you are broadcasting on your LAN, even though your LAN could easily handle, say, the 800 kbps of data your TV card can put out, you may not want to raise the bitrates that high. Having ffmpeg send at, for example, about 640 kbps will help prevent bottlenecks that can occur from slow processors.

When you are streaming video, the less you have happening on the server, the better performance you will get. Running ffserver and ffmpeg doesn't require a GUI. After you run xawtv to select and adjust your video input, you could change to init state 3 (type **init 3** as root); then log out. You will see a simple text login prompt. Log in as root and type the following to start the server and begin streaming:

```
# ffserver &
# ffmpeg -ac 1 -vd /dev/video0 http://localhost:8090/feed1.ffm
```

Improving the streaming video client (mplayer)

If you are getting poor response from mplayer, try adding a `-hardframedrop` to the command line. By dropping more frames, video can actually start to look smoother.

To have some fun with mplayer, try the ASCII art driver (aa) by starting mplayer as follows:

```
# mplayer -vo aa http://localhost:8090/test1.avi
```

Step back from the screen a bit. The aa driver represents the video in text characters. You can actually run mplayer from a shell with no GUI running on your computer.

You can try to use mplayer with other video drivers as well. Type `-vo help` to see what other drivers may be available.

Chuck's Tech Tip

I determined that data was being received from the TV card video source at about 800 kbps. I also know that the human brain perceives smooth motion at around 30 frames per second. Using those two pieces of information, I had an upper bound on the video settings I could use. I found that using a video bitrate as low as 640 kbps resulted in near perfect recreation of the original video on the client machine. At around 512 kbps and lower, the video artifacts started becoming very apparent when the television action got busy.

Summary

Using ffmpeg and ffserver, you can stream video from your Red Hat Linux computer to your LAN or to friends on the Internet. Although the tools I describe in this chapter for serving streaming video on a network are still emerging technologies in Linux, they can work pretty well. You can use this streaming video to monitor video cameras, broadcast a meeting to a few friends or co-workers, or monitor a child's room from another part of the house or from work. The mplayer video/audio player can play your streaming audio and video using the address (URL) of your streaming server and content.

Building a Temperature Monitor

chapter 9

in this chapter

- ☑ Installing temperature monitor software
- ☑ Installing DigiTemp temperature sensors
- ☑ Checking temperature with the digitemp command
- ☑ Adding temperature to a Web page
- ☑ Adding temperature to an e-mail message

You can take one piece of information a long way in Linux. A simple number representing the current temperature can be read into an e-mail, posted on a Web page, or used in dozens of other ways. Gather some temperature readings over time, and you can chart or graph temperature patterns.

The Linux Toys Temperature Monitor comes from Brian Lane and the DigiTemp project (`www.digitemp.com`). The temperature monitor lets you gather temperature data from a simple DigiTemp 1-Wire Sensor. You can purchase fully constructed sensors from the same Web site.

Of course, the real fun is what you do after you have gathered some temperature data. After describing how to get and set up your temperature monitors, this chapter contains several ways of using DigiTemp. These include:

- **Current temperature** — Read the sensors and display the current temperature.
- **Log temperatures (set time)** — Run DigiTemp to get the temperature for a set number of times at set intervals.
- **Log temperatures (cron)** — Use the cron facility to have the system automatically run DigiTemp at selected times and store the output in a log file.
- **Web page** — We include a template for you to publish your own Web page that reads in your temperature data.
- **E-mail** — We describe how to use DigiTemp to automatically add the current time and temperature to the signature of each e-mail message you send.
- **Graph temperatures** — Output your temperature data to a graph.

To gather temperature data, this project incorporates inexpensive temperature sensors that connect together with standard telephone wire and plug into a serial port. We tell you where you can buy the sensors in a kit or put them together from pieces yourself.

Starting Up Your Temperature Monitor

To start up your temperature monitor, you need to connect the temperature sensors to your computer and install Red Hat Linux and the ltweather software. Next, you configure DigiTemp to read temperature information and run scripts to output that information in different ways.

Step 1: Gather the hardware

The only hardware you need for this project is a temperature sensor and a PC. To run the basic DigiTemp software, the PC doesn't require much power. So this is a good project to do on a bare-minimum PC. What you do with the temperature information you gather may require you to have more disk space.

The details of the hardware used for this project are the following.

The PC

You can use the minimum PC that will work with Red Hat Linux for this project. That means that you can get by with a Pentium class (200 MHz minimum) PC with less than a 500MB hard disk. See to Appendix C for the amount of disk space you need if you want to do a Red Hat Linux server install (to add temperature data on a Web server) or a Personal Desktop install (to use any GUI tools with your temperature data).

The only other hardware requirements for the PC itself are:

- **CD-ROM** — You need this to install Red Hat Linux as well as the Linux Toys software.
- **Serial port** — The DigiTemp temperature sensor is made to connect to a serial port on your PC.

Any other hardware you would need on the PC depends entirely on what you want to do with the temperature data you gather. If you are using the data as a temperature server, you probably want to add an Ethernet card.

Temperature sensors

The temperature sensors we used for this project are from the DigiTemp Web site (`www.digitemp.com`). Brian Lane, the maintainer of the project, puts together the sensors in his spare time. If you prefer, you can buy the parts and put them together yourself, if you are handy with a soldering iron.

The DigiTemp 1-wire temperature sensor provides an interface to any of several different digital temperature sensors from Dallas Semiconductor (`www.maxim-ic.com`). These include the DS18S20 (DS18S20-PAR), DS1822 (DS1822-PAR), and DS18B20 (DS18B20-PAR) digital temperature sensors.

For this chapter, we used the DTLT-BASIC kit. The kit contains everything you need to attach a temperature sensor to a serial (COM) port on your computer and to run it a few feet away. Here's what it includes:

- **Serial port adapter (DS9097U)** — 9-pin serial to RJ-11 adapter
- **Cable with RJ-11 connectors** — 6-foot straight-through cable
- **DT1A** — PC board with two connectors and one sensor on it
- **Sensor on a Wire (SOAW)** — A six-inch length of four-conductor telephone cable with a Dallas Semiconductor DS18S20 temperature sensor (with a heatsink over the sensor)

Figure 9-1 shows an example of the DTLT-BASIC temperature sensor kit.

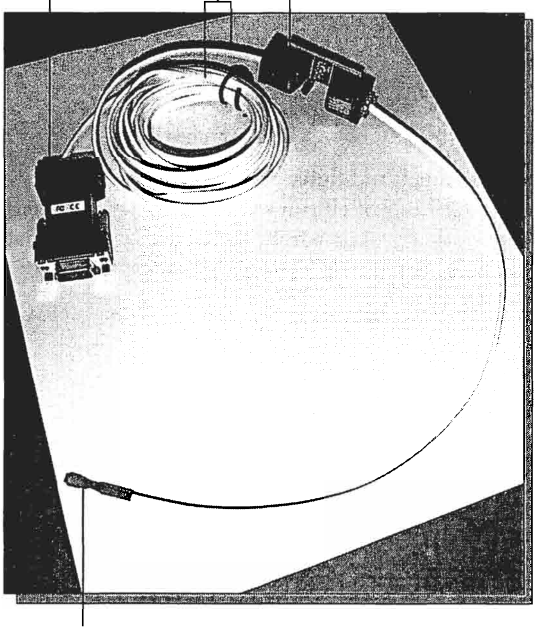


FIGURE 9-1: Connect temperature sensors to your serial port with telephone cable.

Here are a few features of the DLT-BASIC temperature sensor:

- Because each sensor has a unique serial number, you can add multiple sensors to each device. The DigiTemp software will be able to read each individually. The number of sensors you can have is limited by the length and type of cable.
- Temperatures between -55 C and +70 C can be detected. (You can boost the range up to +120 C by supplying +5V to the power pin of the DS18S20.)
- The accuracy of the device is +/-0.5 C (0.9 F).
- You can add a coupler chip (DS2409 from Dallas Semiconductor) to support 1-wire hubs.
- Cable runs of up to 100 meters should be acceptable (although there are reports of cable runs of up to 300 meters). To do runs longer than the 6-foot cable provided, you can add your own straight-through telephone cable. The diode in the device is meant to handle the longer runs.

Note You can refer to the DigiTemp (`www.digitemp.com`) or Dallas Semiconductor (`www.maxim-ic.com`) Web sites for further details on features and other choices of sensors.

Step 2: Install Red Hat Linux

The DigiTemp sofware itself doesn't require much additional software support. In fact, you can start with a minimum install type of Red Hat Linux if all you want to do is read and gather temperature information. (You'll add a few other packages from the Linux Toys CD that aren't in Red Hat Linux.)

To use some of the supporting software, however, there are other Red Hat Linux software packages you should add.

Step 3: Install Linux Toys software

To do this project, you need to install the ltweather RPM package. You can do this as follows:

1. Insert the Linux Toys CD into your computer's CD-ROM drive.
2. As root user from a Terminal window, type:

```
# mount /mnt/cdrom
```

3. To install the package, type:

```
# rpm -Uhv /mnt/cdrom/ch09-DigiTemp/ltweather*
```

Step 4: Attach DigiTemp hardware

Plug the serial port adapter into a COM port on your computer. Cables should already connect the DT1A to the SOAW. Then just run the cable so that the sensor goes to where you want to monitor the temperature.

Each DTLT-BASIC kit comes with two sensors: one on the DT1A board and one on the SOAW. You can use this 1-wire bus as a long line of sensors, placed along the length. You could, however, purchase multiple sensors SOAW and use an RJ-11 splitter (the same thing you would use to split one telephone jack to multiple telephones).

Note

Using a splitter on large networks could result in reflection problems if you are reading sensors on different cable lengths.

As the sensor-on-a-wire temperature sensor is delivered, it is not particularly waterproof. So if you use it outside, you should make provisions to protect it from getting wet.

Here are some practical (and some not-so-practical) places to put your DigiTemp sensors:

- **Your office or home** — Most people will just put the sensor inside or outside of their homes or businesses.
- **Computer room** — Placing a sensor in your computer room can let you know if the room is overheating.
- **Wine cellar** — Make sure that your fine wines are staying properly chilled.
- **Meat freezer** — Check that your side of beef is staying properly frozen.
- **Bat house** — Someone actually did this (`www.batbox.org`) after finding out that the temperature inside a bat house affects whether or not a bat will live there.
- **Wherever** — Any other place where you would want to know if the temperature is too cold, too hot, or just right.

Chuck's Tech Tip

Temperature fluctuations can result in huge financial losses depending on where they occur. The implementation of a device of this nature is a cost-effective method of ensuring such losses do not occur. If you are serious about such a device, consider building it in a small form-factor package such as mini-itx (`http://www.mini-itx.com`) or the embedded PC104 platform (`http://www.pc104.org`).

Step 5: Try the DigiTemp software

To check that your DigiTemp temperature sensors are working, you need to run the `digitemp` command. With your DigiTemp sensor connected, run the following steps (as root user from a Terminal window):

1. Assuming you are connected to the COM1 (ttyS0) port on your computer, run the following commands to create a `.digitemprc` file in the `/usr/local/bin` directory:

```
# cd /usr/local/bin
# ./digitemp -s/dev/ttyS0 -i
DigiTemp v3.2.0 Copyright 1996-2003 by Brian C. Lane
GNU Public License v2.0 - http://www.brianlane.com
Turning off all DS2409 Couplers
..
```

```
Searching the 1-Wire LAN
1025042E000800AA : DS1820 Temperature Sensor
107FC048000800EF : DS1820 Temperature Sensor
ROM #0 : 1025042E000800AA
ROM #1 : 107FC048000800EF
Wrote .digitemprc
```

The ROM numbers are unique to each of your sensors, listed in the order they are found on the wire. When the `digitemp` command is done, it writes the `.digitemprc` file in the current directory (`/usr/local/bin`).

Note Regardless of where the `digitemp` command is, it will look for a `.digitemprc` file in the current directory when the command is run. For example, if you run `digitemp` in a Web page like `/var/www/html/index.php`, a `.digitemprc` file will have to be created in `/var/www/html`.

2. Next, run the `digitemp` command to read temperatures from all (`-a`) the sensors:

```
# digitemp -a
DigiTemp v3.2.0 Copyright 1996-2003 by Brian C. Lane
GNU Public License v2.0 - http://www.brianlane.com
Jun 10 23:50:26 Sensor 0 C: 24.00 F: 75.20
Jun 10 23:50:28 Sensor 1 C: 25.19 F: 77.34
```

This example shows two sensors: one on the little PC board (DT1A) and the other on the end of the wire (the SOAW).

If you get an error when you run this command, here are a few things to try:

- Check that there is a `.digitemprc` file in the current directory. (You will probably get a read/write error if a file is not there.)
- Check that your COM port (COM1, COM2, COM3, and so on) matches the device name (ttyS0, ttyS1, ttyS2, respectively) when you initialized the device.
- Check that permissions are open on the port (`ls -l /dev/ttyS0`). To leave permissions wide open, as root user type: `chmod 666 /dev/ttyS0`

3. If everything seems to be working, you can try out a few `digitemp` command lines. The first example, the `-w` option, "walks" you through all the devices on the wire, so you can check what sensors are on the line:

```
# digitemp -w
Devices on the Main LAN
1025042E000800AA : DS1820 Temperature Sensor
107FC048000800EF : DS1820 Temperature Sensor
```

Here's an example of how to output a line of text, with time and temperature information in it. The following example suppresses the header (`-q`), reads the first sensor (`-t 0`), and outputs text (`-o`) I entered while it reads in the date (`%b %d`), time (`%H:%M:%S`), temperature in Fahrenheit (`%.2F`), and Centigrade (`%.2C`). (The two command lines shown should actually be on the same line.)

```
# digitemp -q -t 0 -o"On %b %d at %H:%M:%S in NYC the \
   temperature is %.2F (F) %.2C (C)"
On Jun 11 at 12:33:41 in NYC the temperature is 75.20 (F) 24.00 (C)
```

The next example suppresses the header (-q) and gathers data twice (-n 2) at intervals of three seconds (-d 3). You can increase the intervals and number of times to gather data to test temperatures over a longer period.

```
# digitemp -q -a -d 3 -n 2
Jun 11 00:26:34 Sensor 0 C: 24.00 F: 75.20
Jun 11 00:26:35 Sensor 1 C: 25.12 F: 77.22
Jun 11 00:26:37 Sensor 0 C: 24.06 F: 75.31
Jun 11 00:26:38 Sensor 1 C: 25.19 F: 77.34
```

Now that you can read the temperature information from your Linux system, you can try out a few interesting ways of using that information.

Logging Temperatures Continuously

If you want to keep track of temperature changes over time, you can use the Linux cron facility to run DigiTemp on an ongoing basis at set intervals to a log file. You can grab data from that log file to read into a Web page, e-mail message, or graph. Here's a procedure for doing that.

1. As root user from a Terminal window, type the following:

   ```
   # crontab -e
   ```

   This opens the root user's crontab file in the `vi` text editor.

2. Assuming you have already created a `.digitemprc` file in the `/usr/local/bin` directory (as described earlier), add the following line to the crontab file you are creating:

   ```
   */30 * * * *  cd /usr/local/bin;./digitemp -a -l/var/log/temperature
   ```

   Note: If you do not know how to use the `vi` text editor, here is exactly what you would type to add the line shown: Type the letter `o` to open a line; type the line shown exactly (you can backspace if necessary), and press the Esc key. Then type `ZZ` to save the changes and close the file. If you get stuck, just type `:q!` to exit the file without saving any changes.

   Make sure you got the number of asterisks, the dot, and the semicolon all entered as shown. This will log the temperature from all sensors to the `/var/log/temperature` file every 30 minutes. You could change the 30 shown previously to any number of minutes to change how often the temperature is checked.

   I set the value to 1 minute just to make sure it is running properly. Then I change it back to 30 later. Continue on to check that the crontab entry you added is working properly.

3. Type the following to view the contents of the temperature file:

   ```
   # cat /var/log/temperature
   ```

   If you used the preceding crontab example, temperature listings should show up within a half hour. If the temperature listings show up, you are done. If not, continue.

4. Check the root user's mailbox to see what went wrong. One way to do that is with the mutt mail reader. As the root user, simply type:

   ```
   # mutt
   ```

   Use the up and down arrows to go to the last message; then press Enter. You should see the cron command line that was run and any errors that may have occurred.

5. To correct any errors in your crontab, edit the crontab file again as follows:

   ```
   # crontab -e
   ```

Chuck's Tech Tip

You can purchase 1-Wire hubs for putting multiple temperature sensors on the same bus, or you can simply hack one together yourself. Unlike Ethernet, collisions aren't an issue for 1-Wire devices. This allows you to use a simple phone-extension splitter or to hack together something similar on a larger scale if you need more than two temperature sensors.

Adding Temperatures to a Web Page

Once you have logged some temperatures, you can have fun putting them into different places. We have a modified script from Brian that grabs the latest temperature listings from the `/var/log/temperatures` file and does some simple formatting to use them in a Web page. The script assumes:

- You have two sensors.
- The first sensor is placed by the computer; the second is in the room.
- You want to read the output into a Web page.

To test this script, I put the sensor on the DT1A board near the fan on my computer and ran the SOAW sensor into the room. The device was plugged into the COM1 port on the same computer I'm using as a Web server.

If you have a different number of sensors, different descriptions of where the sensors are, or want the output file to be read to someplace other than the `/tmp` directory, you can edit the script I'm about to describe.

1. As root user from a Terminal window, type:

   ```
   # /usr/local/bin/my_inc_temp
   ```

   This should create the file `/var/www/cgi_bin/temp_now.inc`.

2. List the contents of the file you just created:

```
# cat /var/www/cgi-bin/temp_now.inc
<HR>
<CENTER>
The current temperatures are:<P>
The computer is 88.81 F (31.56 C)<BR>
The room is a Comfortable 73.06 F (22.81 C)<BR>
Last updated: Jun 11 13:58:03<BR>
</CENTER>
```

As you can see, it created a bit of HTML text that includes:

- A line describing the computer temperature
- A line describing the room temperature (the word Freezing, Chilly, Comfortable, Balmy, or Blistering is displayed as well, depending on the temperature)
- The current date and time

3. If you want to change any of the text that appears in the output, you can edit the `/usr/local/bin/my_inc_temp` file (after you make a backup copy, of course).

4. To have the temperature output regenerated every so often, you can add a cron entry for this command as well. As you did before, type:

```
# crontab -e
```

5. Again, the crontab file opens in the `vi` editor. Add the following line:

```
*/30 * * * *    /usr/local/bin/my_inc_temp
```

If you put this line right after the digitemp line described in the previous section, this script will pick up the new temperature readings every 30 minutes (after they are written to `/var/log/temperature` by DigiTemp). Then it will write the output to `/var/www/cgi-bin/temp_now.inc`.

6. Next, I added the following text to a file named `/var/www/html/index2.php`:

```
<HTML>
<BODY>
<?php
$cmd = 'cat /var/www/cgi-bin/temp_now.inc';
print system($cmd);
?>
</BODY>
</HTML>
```

7. You can open the index2.php file in any Web browser (such as Mozilla). Figure 9-2 shows an example of this file:

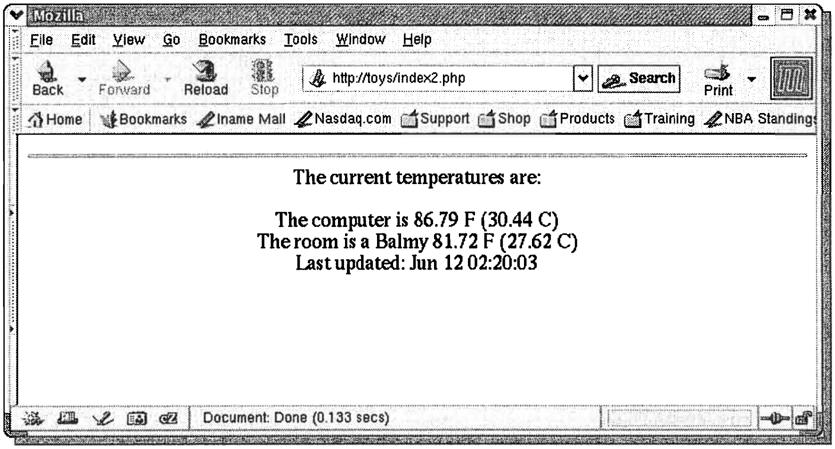


FIGURE 9-2: Show temperature readings on a Web page.

Adding Temperature to Mail Signatures

You can have your temperature output placed in a signature file so it appears at the end of all your outgoing e-mail messages. You can use the `my_sig_temp` script to create that signature file. The `my_sig_temp` script:

- Reads the current temperatures from `/var/log/temperatures`
- Creates a signature file that includes your name and current temperatures

The following procedure lets you create a crontab entry to automatically create and update a signature file with the latest temperatures:

1. Modify the `/usr/local/bin/my_sig_temp` file to change the directory and file to reflect the location of the signature file for your mail reader. For example, if your home directory is `/home/jjones` and you are using Evolution mail, you can change the values as follows:

```
# Directory and filename of signature file to create
$ftp_src_dir  = "/home/jjones/evolution/signatures/";
$ftp_src_file = "signature-0";
```

2. You also want to change the name "John W. Jones" in the my_sig_temp file to your name.

3. Next, add a crontab entry to run the `my_sig_temp` script. As you did before, type:

```
# crontab -e
```

4. Again, the crontab file opens in the vi editor. Add the following line:

```
*/30 * * * *     /usr/local/bin/my_sig_temp
```

As with the Web page script, the my_sig_temp script will pick up the new temperature readings every 30 minutes (after they are written to `/var/log/temperature` by DigiTemp). Then it will write the output to the signature file you indicated.

5. To indicate to Evolution that you want to use a signature file for your outgoing mail, open Evolution (click the mail icon on the panel). Then click Tools → Settings → Mail Accounts (in the left column). Next, click the account name you want and select edit. Click the Add New Signature button. Then select the Unnamed signature.

6. With Evolution still open, select to open a new mail message. The new signature line should appear in the window. Figure 9-3 shows an example of a new e-mail message with a signature that includes DigiTemp temperatures.

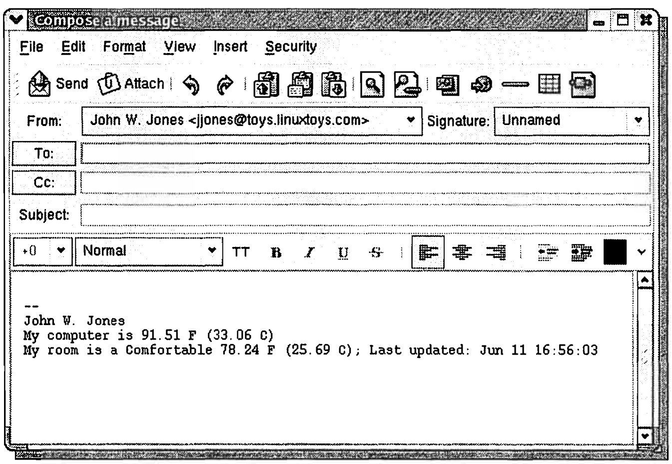


FIGURE 9-3: Have the latest temperature read into your signature line.

More to Do with DigiTemp?

You might want to try a few other scripts that come with DigiTemp. Here are a couple of examples:

- **check_digitemp.pl** — This script, located in the perl directory, is made to be integrated with NetSaint/Nagios monitoring tools. There is information within the script itself on how to install and configure DigiTemp to work with Nagios. To find out more about Nagios, you can visit (`www.nagios.org`).
- **digitemp_sql.pl** — This script lets you log temperatures to a MySQL database. It logs serial numbers and temperature settings in the database. MySQL database software comes with Red Hat Linux. The README file in the DigiTemp source directory describes some of your steps for working with the MySQL database.

To find out more about 1-Wire sensors, here are a few Web sites to check out:

- **Overview of 1-Wire Technology** (www.maxim-ic.com/appnotes.cfm/appnote_number/1796)—Describes how 1-Wire technology works, along with ways of using it
- **Humidity Sensor** (www.ibutton.com/weather/humsensor.pdf)—Tells about the 1-Wire Humidity Sensor
- **Rain Gauge** (www.ibutton.com/weather/1wire_raingauge.pdf)—Tells about the 1-Wire rain gauge.
- **1-Wire Weather Station** (www.sensorsmag.com/articles/0698/wir0698/main.shtml)—Describes the 1-Wire Weather Station
- **Entire 1-Wire product lines** (www.txwx.com/prod.htm)—Weather products from Texas Weather Instruments

Summary

The Linux Toys Temperature Monitor describes how to use DigiTemp hardware and software to gather temperature data and use it in a variety of ways. Temperature information can be read into Web pages, e-mail signatures, or graphs. Using DigiTemp sensors, you can monitor the temperature anywhere within 100 meters or more from your computer.

Setting Up a Digital Receptionist

chapter 10

in this chapter

- ☑ Choosing hardware
- ☑ Installing Red Hat Linux and Toys software
- ☑ Setting up the modem
- ☑ Listening for calls on the modem
- ☑ Configuring voice recordings
- ☑ Creating basic voicemail settings
- ☑ Planning your voicemail
- ☑ Recording incoming voice messages
- ☑ Checking your e-mail for voice messages

If you're addicted to e-mail (*I can read it when I want*), but you find the telephone annoying (*it's always trying to interrupt me*), we might be able to help you. The Linux Toys Digital Receptionist project will:

- **Take phone messages** — You can set up your Linux computer to pick up the telephone after a set number of rings and give callers the choice to leave messages to the voicemail box they choose.
- **Send recorded messages** — Take recorded messages and mail them to any e-mail account you choose.
- **Play back messages** — From any e-mail reader, you can open the e-mail message and launch a sound/music player to play back the recorded message on your speakers.

The Linux Toys Digital Receptionist is a repackaging of the VOCP project. VOCP is a very cool voice-messaging system that runs in Linux and other UNIX-like systems. You can learn more about VOCP at `www.vocpsystem.com`.

Besides a PC, the primary piece of hardware you need for this project is a modem that can handle voice (we recommend one and list a few others). Add a phone line, a couple of speakers, a microphone, and you're there.

On the software end, you need to install the VOCP software, configure the message recorder, and create a voice box tree file to configure how messages are gathered and distributed. You have a lot of flexibility in setting up how your messages are handled.

If you play your cards right, you could replace your receptionist (who didn't really like to answer the phone anyway).

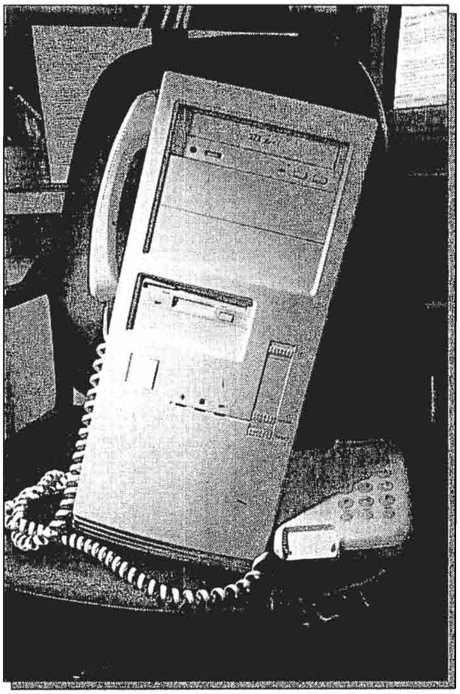


FIGURE 10-1: VOCP may take phone messages better than a receptionist does.

Creating the Digital Receptionist

The main components of VOCP, the open source project on which the Linux Toys Digital Receptionist is based, are:

- **Vgetty** — A modified version of the vgetty daemon process is used to listen for incoming calls and hand them off to be handled by VOCP.
- **Perl scripts** — Most of the VOCP project consists of perl scripts that are used to create and modify the recordings and configuration files you need to set up your VOCP answering machine.

Our project uses only some of the features of VOCP. I encourage you to look for more information and help from other people using this software from `www.vocpsystem.com`. Besides converting phone messages to e-mail, VOCP can be used to:

- View and send faxes (using VOCPhax)
- Translate e-mail text to speech (TTS). (This feature relies on the festival project, which you can learn about at `www.cstr.ed.ac.uk/projects/festival`.)
- Use caller ID redirection and filtering

In our project, we are using the Ogg Vorbis audio format to compress the answering machine messages we send. Check out `email-attachments.txt` file in `/usr/local/vocp/doc` for information on using ogg and possibly replacing it with MP3 voicemail attachments. (There is also other useful information in that `doc` directory.)

Step 1: Gather the hardware

As I noted earlier, the primary hardware you need with this project is a PC and a supported voice modem. A few extra pieces will be helpful as well, such as some speakers, a microphone, and a network card (since you need to send the e-mail). And, of course, you need a telephone line to answer.

The personal computer

The speed of the PC's bus is not as critical with audio projects, such as this one, as it is for video projects. Most PCs that can run Red Hat Linux will work with this project.

Hard disk

Your computer will need at least 1.7GB of disk space to do a Red Hat Linux Personal Desktop install. You need less than 10MB for the Linux Toys packages you need for this project. Other than that, each voicemail message will take only a few kilobytes each of disk space.

CD drive or floppy disk drive

You need some medium for installing Red Hat Linux and Linux Toys software. CD is the preferred medium, but a floppy disk and an Ethernet network card will work as well.

Network card (recommended)

Unless you are just storing messages locally or e-mailing them to users on the local computer, you need a network card (or at least a modem), so that the e-mail messages can be sent to users off of the voice recorder machine.

Modem

This project requires a modem that supports voice. Don't just pull any old modem out of the bin at the local computer store and expect it to work. We used the following modem with this project and found it to work quite well:

- MultiTech MultiModem MT5600ZDXV External

Chuck tried a few other modems as well, but the MT5600ZDXV worked the best. We had some problems with low volume levels when messages were played to callers through the modem, but we were able to overcome that. The VOCP project also lists the following modems as being reported to work:

- Zyxel 1496 and 2864
- MultiTech MultiModem MT2834 ZDX V
- ELSA MicroLink TQV series
- US Robotics/3Com 56K Voice FaxModem Model 5610
- Diamond 56e Pro voice

- External Zoom/Faxmodem V.92
- Zoom 56K PCI internal

Attach the external modem to a serial port (COM) on your computer. When Red Hat Linux is installed and booted, your modem should be automatically detected and linked to the `/dev/modem` device.

Chuck's Tech Tip

By far the hardest part of this project was in finding a modem that actually works. In the fax modem world, research has shown that almost no modem firmware does FAX 2.0 correctly. The conventional wisdom with projects like Hylafax is to use your modem in fax mode 1.0 and use software to do the proper mode 2.0 translation. This is where I believe winmodems are a blessing in disguise. A winmodem offloads most of the processing work to the machine through a software driver. This would enable a driver to be written that implements voice capabilities properly. As of the time of this writing, no such driver exists.

Speakers

It's handy to have two sets of PC speakers when you set up this project. Attach one set to the sound card so you can check and play back messages. Attach the other set to the speaker jack on the modem so you can test raw data messages playing on your modem.

Sound card

You need a sound card to plug in your microphone and speakers and to record and play back messages, respectively. Choosing and configuring sound cards is discussed in Chapter 3.

Microphone

You need a microphone to record the messages that are played when people call your message machine. Many PCs come with microphones that either plug into the sound card or are built into the PC.

A telephone line (and an extra telephone for testing)

You need a mobile phone, a second phone in your house, or neighbor's phone to be able to call your voice modem and test it out. You won't need the extra phone once you're done setting up the project.

Step 2: Install and configure Red Hat Linux

When you install Red Hat Linux, you can do a Personal Desktop install. You really need only the GUI, however, if you want to use the graphical window VOCP provides for setting up voice boxes.

Cross-Reference

See Appendix C for information on installing Red Hat Linux.

Step 3: Connect the voice modem

Here are a few steps you need to run to set up the MultiTech MultiModem (MT5600ZDXV). The procedure may vary if you have a different modem:

1. Connect the modem to any COM port on your computer. If you use COM1, the device name you need to know later is `/dev/ttyS0`. COM2 is `/det/ttyS1` (you get it).
2. Connect speakers to the speaker jack.
3. You don't need to connect a microphone to the microphone jack. (You should, however, have a mic handy to connect to your sound card later to record your voicemail messages.)
4. Plug the phone line into the line jack.
5. Plug in the power.

Figure 10-2 shows an example of the connections for the MT5600ZDXV we used to build this project.

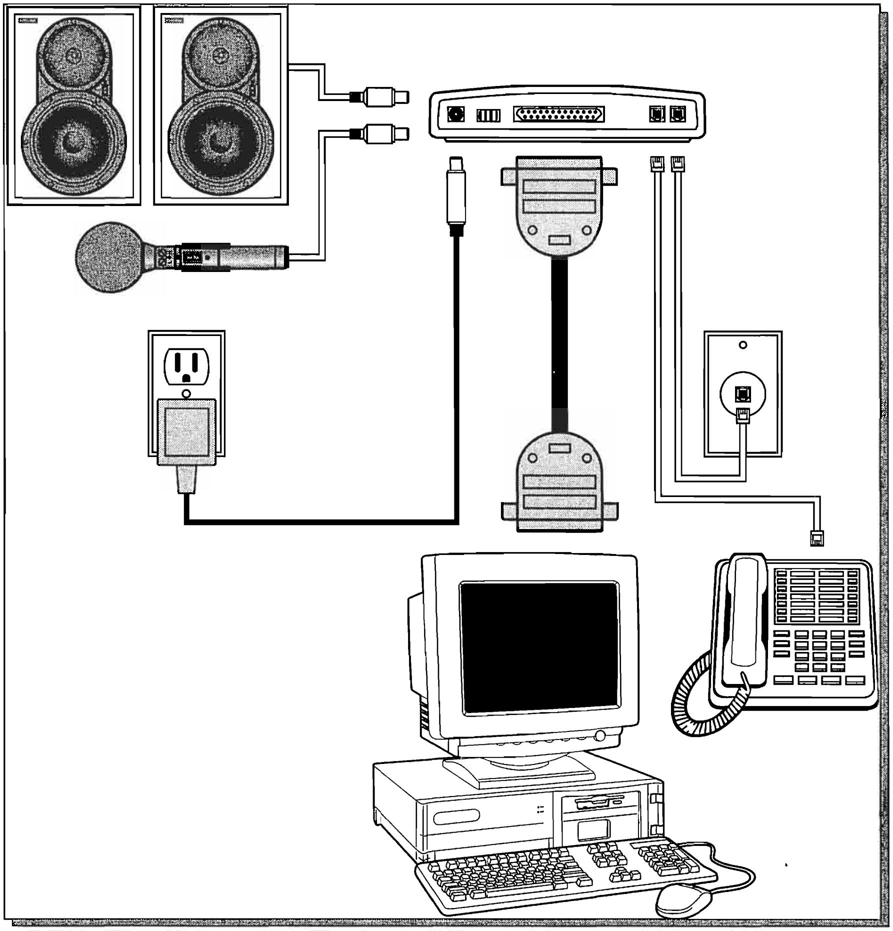


FIGURE 10-2: Connect your voice modem to a serial port, speakers, phone line, and a phone.

Step 4: Install Linux Toys packages

The Linux Toys Digital Receptionist consists mainly of the vgetty daemon (listens on a serial port for incoming calls to the modem) and a set of perl scripts that configure and implement the project. Follow this procedure to install that software:

Note

To use MP3 encoding, VOCP expects you to have the lame package installed. If you install lame (it is included in our packages for Chapters 4 and 5) in addition to the packages named here, you can set the recorder to create messages in MP3 format.

By default, we use the `oggenc` command (which comes with Red Hat Linux) to do encoding to Ogg Vorbis audio format. Because Ogg Vorbis is less common than MP3, if you don't switch to MP3, you may have to add an Ogg Vorbis player to the computer on which you receive your e-mail. We describe where to get those clients later in this chapter.

1. With the voice modem connected, restart the computer. As Red Hat Linux boots up, the modem should be detected. If prompted, go ahead and configure the modem.
2. Log in as root user.
3. Remove any mgetty RPMs that are installed on your computer. To make sure that is done, type the following:

   ```
   # rpm -e mgetty-voice mgetty-viewfax mgetty-sendfax mgetty
   ```

4. Insert the Linux Toys CD, mount it, and install the packages needed for this project as follows:

   ```
   # mount /mnt/cdrom
   # cd /mnt/cdrom/ch10-DigitalReceptionist
   # ./install.me
   ```

The VOCP software is now installed and ready to be configured.

Step 5: Configure vgetty to listen for calls

The vgetty daemon is used to listen for incoming calls on your phone line. The following section describes how to configure the special patched vgetty included with this project:

1. Log in as the root user.
2. Open `/usr/local/etc/mgetty+sendfax/voice.conf` (with any text editor), and check the values in that file. We expect that the defaults in this file will work in most cases. Here are a few particular values you might want to look at:

   ```
   voice_devices modem
   rings 3
   call_program /usr/local/vocp/bin/vocp.pl
   ```

When you booted Red Hat Linux after installing the modem, it should have automatically linked your modem (`/dev/ttyS0` for the COM1 port) to the modem device (`/dev/modem`). You can type `ls -l /dev/modem` to make sure that your modem is properly linked to `/dev/modem`.

The default number of rings before the modem picks up is three (`rings 3`). I changed this value to 2, temporarily, while I tested a few incoming calls. Then I changed it back to 3.

When a call is received, the value set for call_program (the default is `vocp.pl`) is run to handle interactions between VOCP and callers. If you can read perl, it might be interesting for you to see how this script behaves.

3. Open the `/etc/inittab` file (as root user with any text editor), and check that the following lines appear at the end of the file (installing VOCP automatically creates these lines):

   ```
   # Vgetty respawn line.
   SX:345:respawn:/usr/local/sbin/vgetty modem
   ```

   Again, you might want to check that your modem is linked to `/dev/modem`. (For example, typing **ls -l /dev/modem** should show the device linked to `/dev/ttyS0` if your modem is connected to the COM1 port.)

4. Start the `vgetty` process by having `init` reread the `inittab` file as follows:

   ```
   # init q
   ```

5. If everything worked right, the `vgetty` process should be running. Type the following to make sure:

   ```
   # ps -ef |grep vgetty
    5669  ?      S    0:00 /usr/local/sbin/vgetty ttyS0
   ```

6. If `vgetty` is running, open another Terminal window and begin to tail the output of the log file by typing the following (replacing ttyS0 with the COM device you are using):

```
# tail -f /var/log/vgetty.modem
    .
    .
    .
07/09 15:34:25 dem  detecting voice modem type
07/09 15:34:26 dem  Multitech_5600ZDXv detected
07/09 15:34:27 dem  initializing Multitech MT5600ZDXv voice modem
07/09 15:34:29 dem  waiting...
07/09 16:34:29 dem  checking if modem is still alive
07/09 16:34:29 dem  mdm_send: 'AT' -> OK
07/09 16:34:29 dem  waiting...
```

   Leave the process running in the Terminal window while you go on to test the project.

Step 6: Configure modem compression

To configure voice recording, you need to identify your modem, change the sample rate, and try playing a few messages. Before you begin, make sure that speakers are connected to your modem's speaker port (if it has one). To detect your modem and convert the sample rate on your outgoing messages, run the vgetty-test.pl script as follows:

1. Log in as root user.
2. From a Terminal window, type the `vgetty.pl` command as follows:

   ```
   # /usr/local/bin/vgetty-test.pl
   Enter full path to vgetty voice.conf file
   [/usr/local/etc/mgetty+sendfax/voice.conf]:
   ```

3. Press Enter to accept the location of the voice.conf file being used. You are prompted for the modem to use:

   ```
   Enter modem device (ttyS0 for '/dev/ttyS0)
   [modem]:
   ```

4. Press Enter to accept modem (which represents `/dev/modem`) as the device to which your voice modem is connected. You are prompted to select where the sound will be played when you try to play voice files to test your modem:

   ```
   Select output for vm command. Possible values are:
   Enter selection [2]: 6
   ```

5. We selected 6 for internal speaker (even though it played on our external modem speakers). If you don't hear anything during the test that follows, you can run the command again and try a different output location. Next, the script begins testing for each type of modem (actually chipset and compression level) it knows about:

   ```
   Running vm play for Digi 4 - hit ^C (Ctrl-C) once to abort
   Could you correctly hear the test sound file? [n]
   ```

6. If you did not hear the sound, type **n** and it will continue to test other modem types. If you heard the sound clearly (it's Kevin offering the proper pronunciation of Linux Toys), type **y**. For the MultiTech 5600ZDXv modem we used, there was a MT_5634 selection that worked, for either Rockwell 2 or Rockwell 4. (We chose Rockwell 4.)
7. When you finally find a working modem setting (and press `y`), VOCP changes the configuration file to use your settings. It also converts all the message files (in `/var/spool/voice/messages` and its subdirectories) to use the proper compression. The settings used to convert the messages to the proper modem type and compression rate are displayed when the script exits. Here's an example:

   ```
   ltdrconfig Rockwell 4 7200=
   ```

   When we create our own voice messages later, we can convert them to the proper format using the `ltdrconfig` with the options shown here. (For a different modem, you will see different settings.)

8. To make sure that the messages were translated properly, go to the messages directory and play a message. Here's how:

```
# cd /var/spool/voice/messages
# vm play -s -v english.rmd
```

The voice message should play on the modem's internal speakers or the external speaker connected to the modem. Then you should see the modem reset (from tailing the log file as we did a few steps earlier). For your modem, you may need to set the output device with `-e` (for external speakers) instead of `-s` (for internal speakers).

9. Next, test that the modem is able to pick up and play back a recorded message. Dial the number of the phone line your modem is connected to (use your cell phone or second phone in your house). You should hear the welcome message (`root.rmd` file), followed by a list of options you have for selecting different ways of leaving a message. If you hear the message, hang up and go ahead to "Step 7: Set general compression and e-mail settings."

10. If you have trouble getting the voice messages to be loud enough for the person who has dialed into the modem (as we did), you can run the `ltdrmaxvolume` command. This command tries to use the options determined for your modem to increase the volume of all files in the `/var/spool/voice/messages` directory and its subdirectories. If you like, you can insert those values yourself (from the output you saw in an earlier step). For example:

```
# ltdrmaxvol Rockwell 4 7200
```

Step 7: Set general compression and e-mail settings

The VOCP general configuration file (`/etc/vocp/vocp.conf`) contains the basic settings needed to do voice recording and to manage how the messages are distributed. The most important options you need to set in this file include:

- **Modem and compression** — Options that set how the particular modem you are using handles compression
- **E-mail notification** — Options to determine how recorded phone messages are sent in e-mail

There are many other options you can consider changing in this file. I list some of those at the end of this step. Open the `/etc/vocp/vocp.conf` file (as root, using any text editor). Then begin going through the options in the sections that follow:

- **Voice box tree** — The boxconfig option sets the location of the voice box tree. By default, its location is `/etc/vocp/boxes.conf` as follows, which you shouldn't change. (You edit the `boxes.conf` file in a later step to set up your voicemail boxes.)

```
boxconfig     /etc/vocp/boxes.conf
```

- **Raw Modem Data Format (rmd)** — It's important to get these settings right. The rmdformat and rmdcompression options were changed automatically during the modem detection earlier to Rockwell and 4, respectively, for the MultiTech MT5600ZDXV modem we used. (Type **pvftormd -L** to see other format and compressions, if you have a different modem.) The rmdsample was set to 7200.

  Defaults should be fine for the next few options. Make sure that the VOCP commands are in /usr/local/bin (as indicated by pvftooldir). The pause option sets the number of seconds (3) to wait for user input. The max_errors value sets how many invalid selections a user can make in a mailbox before being disconnected (2 by default).

```
rmdformat          Rockwell
rmdcompression     4
rmdsample          7200
pvftooldir         /usr/local/bin
pause              3
max_errors         2
```

- **E-mail notification** — Here you set whether or not to have the audio phone voice message sent with the e-mail notification. You also set how you would like that message compressed (Ogg or MP3).

  Setting email_attach_message to 1 causes the actual audio message to be attached to the e-mail notifying you that a phone message arrived. Set it to 0 if you want to just get a notice; then play the message later from Linux. I changed it to 1 to have the audio message attached.

  By default, messages are compressed (and later played back) using Ogg Vorbis audio compression (message_in_email_format ogg). You can set it to mp3 instead.

Note

When VOCP encodes its recorded phone messages, it uses scripts in the /usr/local/vocp/bin directory. For Ogg it runs pvftoogg; for MP3 it runs pfvtomp3. Those two scripts do encoding with the oggenc and lame commands, respectively.

The other options that follow set sendmail as the program to send your e-mail. Next, email_subject sets the subject for pager boxes and e-mail boxes to "VOCP VoiceMail" and email_from_address sets the From: field on the messages sent to vocp@localhost.localdomain. You probably want to at least change the email_from_address. You could change email_subject to something like "Jones Family Voicemail" instead.

```
email_attach_message      1
message_in_email_format   ogg
programs                  email    /usr/sbin/sendmail
email_subject             "VOCP VoiceMail"
email_from_address         vocp@localhost.localdomain
```

Chuck's Tech Tip

Ogg Vorbis uses a variable bitrate encoding scheme for optimal sound quality while minimizing disk utilization. When considering a sound encoding scheme, consider technical features as well as the fact that Ogg Vorbis is unencumbered by licensing restrictions. The group that owns the rights to the MP3 encoding format recently started enforcing licensing restrictions that have resulted in a scarcity of MP3 players delivered with Red Hat Linux. For more information on MP3 licensing, see `http://www.mp3licensing.com`.

- **Other options** — In most cases, you don't need to change most of the other options available in the `vocp.conf` file. Here's how other options that may interest you are set in this file:

  ```
  call_logfile              /var/log/vocp-calls.log
  log_incoming              1
  default_branch_to         0
  repeat_message_on_error   1
  ```

  If you need to debug your VOCP setup, as set by `call_logfile`, log messages are sent to the `/var/log/vocp-calls.log` file. Having `log_incoming` set to `1`, causes all incoming calls to be logged. With `default_branch_to` set to `0`, if the user does not enter any input, VOCP will not branch to another box. (Change this to a branch number to have it branch to a particular box.)

  If a user enters an invalid entry, with `repeat_message_on_error` set to 1, the original message is repeated (if 0, the message isn't repeated).

Note

If your phone service offers caller ID, you can make your message recorder behave differently based on caller ID. For example, you could have incoming calls from certain numbers jump immediately to certain message boxes. For further information on using caller ID with VOCP, refer to `/usr/local/vocp/doc/caller-id.txt`.

Step 8: Plan your voicemail system

A good way to learn how to make your personal voicemail system is to start with the default voicemail boxes that are in place. The voicemail boxes that are delivered with VOCP are configured in the `/etc/vocp/boxes.conf` file.

The `boxes.conf` file is in XML format. To simplify viewing the file and changing its contents, VOCP comes with the VOCP boxconf window. I'll step you through using the VOCP boxconf window to edit the `boxes.conf` file a bit later.

Figure 10-3 shows a typical path that a call might take, based on the default boxes that are set up in the `boxes.conf` file with VOCP.

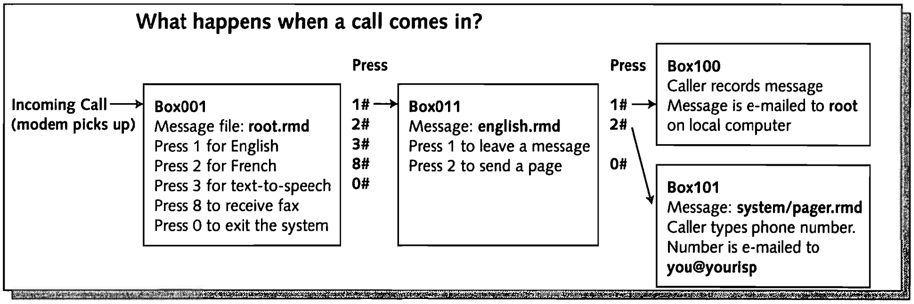


FIGURE 10-3: When a call comes in, each box determines what happens next.

Here's what happened when a call came in to the voice modem, using the default `boxes.conf` file:

- The modem picks up the incoming call.
- VOCP box 001 plays the `root.rmd` file (from `/var/spool/voice/messages`).
- The caller presses 1# and is directed to Box 011. (The caller could also have pressed 2# to get a message in French, 3# to use text-to-speech features, 8# to have the modem receive a fax, or 0 to exit and hang up.)
- When the caller presses 1#, Box 011 is selected and the `english.rmd` file is played.
- The caller can select 1 to leave a message after the tone or 2 to type a phone number to send.
- If the caller presses 1#, Box 100 is selected, allowing the caller to leave a voice message that is recorded and e-mailed to root user. If the caller presses #2, Box 101 is selected. The `system/pager.rmd` message is played, and the user is prompted to touch her phone number into her telephone. That phone number is e-mailed to *`you@yourisp`* (that obviously needs to be changed).
- Then either the recorded message or the phone number the caller entered is e-mailed to the e-mail address associated with that box.

To create a personal voicemail system for yourself, you can modify the `boxes.conf` file and record a few of your own messages. Here's a quick procedure for doing that:

1. Make a copy of the `boxes.conf` file (as root user from a Terminal window):

   ```
   # cp /etc/vocp/boxes.conf /etc/vocp/boxes.conf.original
   ```

2. Go to the VOCP bin directory and open the VOCP boxconf window:

   ```
   # cd /usr/local/vocp/bin
   # ./boxconf.pl
   ```

   The VOCP boxconf window should appear, as shown in Figure 10-4.

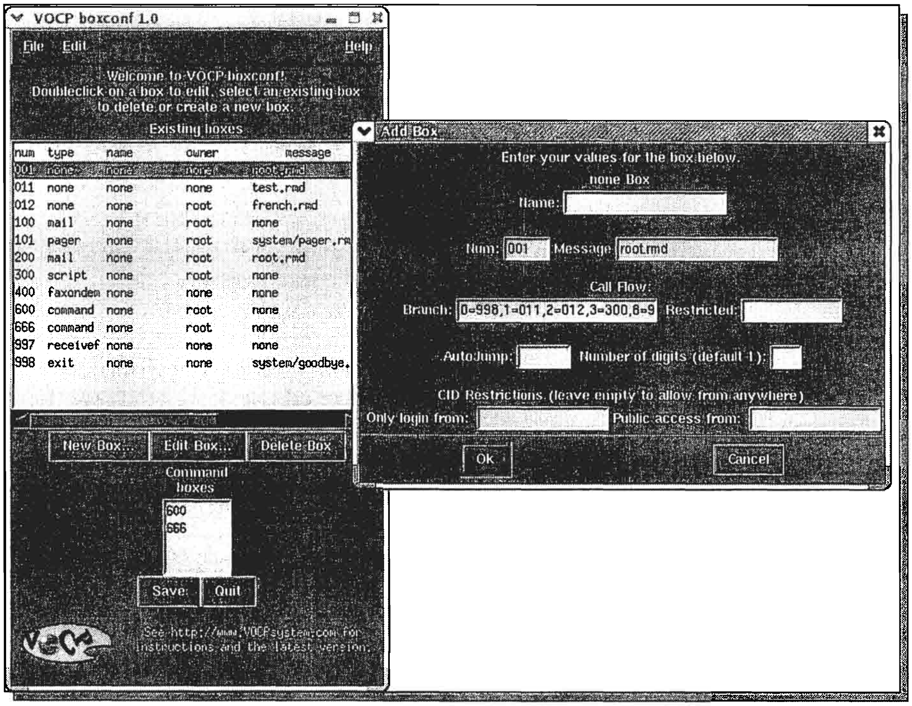


FIGURE 10-4: Create your voicemail boxes using the VOCP boxconf window.

3. Double-click voice box number 001 (`root.rmd`). A window opens that lets you change the root voice box to suit your needs.
4. Rename the message from `root.rmd` to `myvbox.rmd` and click OK. (You will create the `myvbox.rmd` file in the next major step.)
5. Double-click voice box number 100 (`mail`). Again, an edit window opens to let you change the values for this box.
6. Change Email to the e-mail address you would like to use to get voicemail messages and click OK. (I changed it from `root` to `chris@linuxtoys.net`.)
7. Double-click voice box number 101 (pager). An edit window opens.
8. Change Email to the e-mail address you would like to use to get voicemail messages and click OK. (I changed `you@yourisp` to `chris@linuxtoys.net`.)
9. Click Save and, when prompted to replace `/etc/vocp/boxes.conf`, click OK. (Remember you have a backup copy, `/etc/vocp/boxes.conf.original`, so you can always copy that one back to `boxes.conf` if you make a mistake.)
10. If one does not already exist, create a directory to hold incoming messages:

```
# mkdir /var/spool/voice/incoming
```

At this point, you should have a simple voicemail service that will:

- Play your own message (`myvbox.rmd`, which we'll create in the next step) that tells callers how to proceed
- Direct callers to Box011, which lets them either leave a voicemail message (by pressing 1#) or key a phone number (by pressing 2#)
- Send the e-mail message to the e-mail address you selected in the previous procedure. (If you selected to have `email_attach_message` set to `one`, as suggested in an earlier procedure, the audio message is sent with the mail message.)

After you have had some success modifying `boxes.conf` to go to a single person's voicemail or pager (actually, just a phone number in a mail message), you can change the configuration further. You could add your own French, Spanish, or other language message (right now, French is programmed to the 2# key). Or you could add a new mail message to each member of your family, each pointing to boxes that e-mail their messages directly to their e-mail accounts. (For now, however, let's just get one person's voicemail working.)

Step 9: Record your new voicemail messages

Now that you know what messages you want to use in your voicemail system, go ahead and create those messages and convert them to rmd format. Here's how:

1. Plug a microphone into the microphone jack on your sound card.
2. From a Terminal window, type **aumix** to open the sound mixer. With the aumix screen displayed, do the following:
   - Turn the microphone volume all the way up by clicking the slider next to Mic, so that the "o" appears all the way to the right.
   - Click the green `P` in the microphone column so that it turns into a red `R`. This indicates that you will record from the microphone device.
   - Press `S` (to save the settings) and `Q` (to quit `aumix`).
3. Change to the messages directory as follows:

   ```
   # cd /var/spool/voice/messages
   ```
4. When you are ready to record (that is, the microphone is right in your face), type the following command and immediately begin recording your messages. The message could say something like: "You have reached the Jones residence. To leave a message, press 1, followed by the pound sign."

   ```
   # rec myvbox.wav
   ```

   Press Ctrl+C when you are finished recording.
5. Convert the wav file to pvf format as follows:

   ```
   # wavtopvf myvbox.wav myvbox.pvf
   ```

6. Change the sample rate from 8000 to 7200 (or whatever rate your modem supports):

```
# pvfspeed -s 7200 myvbox.pvf myvbox2.pvf
```

7. Next, you want to convert the pvf file to an rmd file as follows (if you have a different modem than we have, use your own settings instead of Rockwell 4):

```
# pvftormd Rockwell 4 myvbox2.pvf myvbox.rmd
```

8. With speakers still connected to your modem, make sure that you can play the rmd file you just created through the modem, as follows:

```
# vm play -s -v myvbox.rmd
```

9. Repeat this procedure for every message that you need for the voice box setup you created in the previous major step.
10. Once your messages are recorded, run the `ltdrmaxvol` command again, as shown earlier, to increase the volume of the messages you just created.

If you set up everything correctly, you should be able to check your new voicemail system and hear your new message, record an incoming message, and have that message sent to the e-mail account of your choosing.

Step 10: Check that your voicemail is working

Call your phone number (this is where you'll need that mobile phone or extra line). Here's what should happen:

1. The modem should pick up the call after the number of rings you set.
2. It should play the message you recorded.
3. When you hear the message, type **1** as instructed.
4. When instructed, type either **1#** to record a message or **2#** to key a phone number.
5. Hang up when you are done recording. The message should be sent.

Step 11: Retrieve voicemail messages (from your e-mail)

If everything went well (let's cross our fingers), you should be able to retrieve your voicemail in your e-mail reader.

Open your mail reader for the e-mail account that you assigned to the message you just sent. I used Evolution e-mail in Linux to open my mail. Figure 10-5 shows an e-mail containing the audio voicemail message. Figure 10-6 shows a message that includes a phone number that the caller keyed in.

With the pager number, you can simply call back the phone number shown. In Evolution mail in Linux, you can simply click on the attachment box and either save the message (message.ogg) to a file or open it using the xmms player. The file will also play in the `ogg123` player.

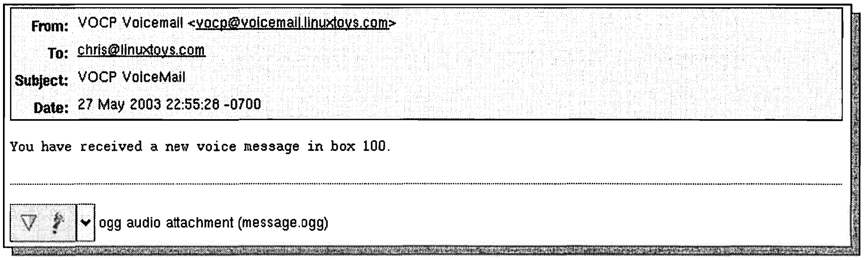


FIGURE 10-5: An audio message (in Ogg format) is attached to the e-mail.

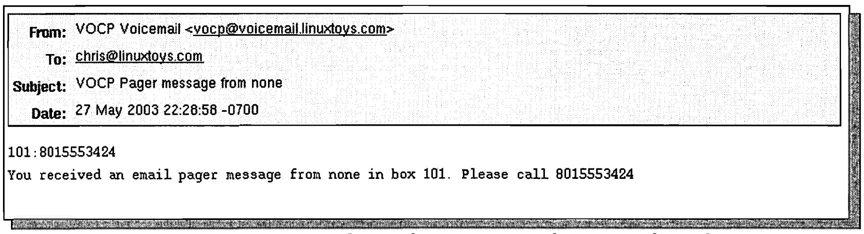


FIGURE 10-6: A pager message results in the incoming phone number's being sent.

If your e-mail reader is on a Windows system, you need get an audio player that can play files in OGG format. Here are two suggestions:

- **Quintessential Player** (`www.quinnware.com/news.html`)
- **Winamp3** (`www.winamp.com`)

Troubleshooting VOCP

The biggest problem we encountered with this project was finding a modem that was fully supported. Messages from the answering machine were nearly inaudible at first with the modem we used. By increasing the volume of our messages, we were able to work around the problem.

If you run into some trouble with this project, here are a few things you can try:

- **Increase debug levels** — Open the `/usr/local/etc/mgetty+sendfax/voice.conf` file and increase the debug level (for example, voice_log_level 6). Try calling into the modem and watching the messages sent to log files in the `/var/log` directory.

  In the `/var/log/vocp.log` file, you can watch all the instructions in each perl script being executed. The `voicelog` file shows the interactions going on between VOCP and the caller. The `vgetty.modem` file will show you which configuration files are being read and whether or not the modem is being initialized properly. In the `vm.log` file, you can see which message files are being played. The `vocp-calls.log` shows calls being logged.

- **Supporting Docs** — For supporting documentation, check the `/usr/local/vocp/doc` directory.
- **Mailing list** — If you get stuck on something, check out the VOCP mailing list: `http://www.vocpsystem.com/maillist.php?mode=component`.
- **Troubleshooting the modem** — Type `minicom -s` to initialize the modem. Try a few AT commands (see the modem's manual) to make sure that it is functioning properly. For the MultiTech MT5600XDXv we recommend, you can find a manual containing AT codes on MultiTech's Web site at `www.multitech.com/DOCUMENTS/manuals/S0000098.pdf`.

Summary

The VOCP project offers a lot of flexibility in setting up voicemail boxes for your Linux Toys Digital Receptionist. Using definitions for different boxes, callers can navigate your voicemail system and record messages or key in return phone numbers. Those audio messages can be attached and sent with e-mail messages, so the recipient of the messages can play them back on a PC.

The trick to this project is getting a supported modem and then creating your own custom messages and configuring the voicemail boxes in a way that callers can navigate your system. If you get stuck, check your log files and connect with the VOCP mailing list.

Small Business Opportunities

part IV

in this part

Be a Mini ISP

chapter 11

in this chapter

- ☑ Learning what's in the Mini ISP
- ☑ Setting up network interfaces and modems
- ☑ Configuring DNS, routing, and firewalls
- ☑ Adding mail, Web, and FTP services
- ☑ Using the Mini ISP

Did you know that all the basic software you need to be an Internet Service Provider (ISP) is right in Red Hat Linux (and most other distributions of Linux for that matter)? You can set up Linux to allow dial-in modems and routing to the Internet, as well as offer Web publishing, e-mail, and file transfer (FTP or other services).

The challenges of being a real ISP come from issues of service, support, maintenance, billing, and reliability. But, if you're willing to stuff all that and just build a low-cost, dial-in-ISP-type service to share with a couple of friends, family members, or co-workers, we've got a nice project for you. When you are done creating your Mini ISP, here is what you will have:

A dial-in server (with 33.6 Kbps modems) that lets your clients:

- Connect to the Internet.
- Publish personal Web pages.
- Upload files to be stored or published.
- Get their personal e-mail.

A public Internet server that lets you:

- Have your own domain name.
- Create a public e-mail server.
- Create a public Web server (described in detail in Chapter 14).

While the server we describe may not help you launch the business opportunity of a lifetime, it will let you learn some of the basic tools used by ISPs. You will also end up with a useful server that you can:

- Dial into while you are on the road.
- Use to allow your out-of-work brother-in-law a way to have an e-mail account (and a way to get to it).
- Give a nonprofit organization a cheap way to get online.

Who knows? Your dial-in modem and PC could help get your friends who are otherwise disconnected get connected to the Internet. (See Figure 11-1.)

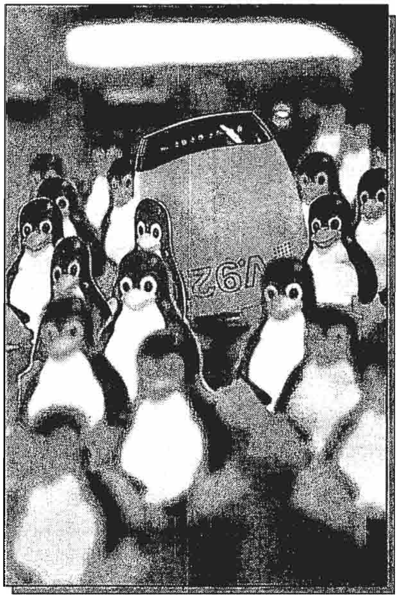


FIGURE 11-1: Get an old PC with a couple of modems and you can help your friends get connected to the Internet.

Caution

On a serious note, this project illustrates how to do a set of features on a single computer that, for security and performance reasons, are typically done on multiple computers. Also, while these procedures are legitimate for setting up the individual services we cover, you should take more care in building Internet servers that protect critical data.

In *Red Hat Linux Bible*, I go into much more detail related to the configuration and security of Linux Internet servers. There are also many other books and Web sites that can help you set up secure Linux servers when you are protecting more than just a few personal Web pages.

If service and reliability are an issue, you can consider hiring someone to maintain your server. For example, a company like Chuck's Quantum Linux Laboratories (`www.quantumlinux.com`) can give support for your Red Hat Linux server; a company like RackSpace (`www.rackspace.com`) can manage a computer for you at its data center.

About the Mini ISP

The Linux Toys Mini ISP project spans a range of features built into Red Hat Linux that we're going to jam into one computer. It consists of a set of canned configuration files and the instructions that come with this book. Besides that, you don't need much more than a PC, a couple of modems (and related phone lines), and whatever equipment you need for a high-speed Internet connection.

While this project is basically an exercise in setting up a range of network services, it can also be useful in situations where low-volume, dial-up Internet access is needed. The concepts of

restricting users' access to their own home directories, opening firewalls to allow network services, and doing simple routing and network-address translation are concepts that you will be able to use as you move further along your journey with Linux.

Configuring the Mini ISP

The general steps for configuring your Mini ISP are as follows:

1. Gather computer hardware.
2. Install Red Hat Linux.
3. Install Linux Toys software.
4. Configure the network.
5. Configure the domain.
6. Configure the Web server.
7. Configure the FTP server.
8. Configure the mail server.
9. Add user accounts.
10. Open the firewall.

Step 1: Gather computer hardware

For the Mini ISP, you need a computer that has enough disk space and processing power to support the people you have using it. Additional hardware items you need include modems, communications hardware to connect to your ISP, and the ability to back up data.

The personal computer

Requirements for performing a Red Hat Linux Server installation set the bottom limit on what will work for this project. Besides that, you need to take into account the resources you want to provide for each user.

Hard disk

Here are some guidelines for choosing the hard-disk size you need for this project.

- **Red Hat Linux** — If you are comfortable working from the command line, you can get by with a Red Hat Linux server install, requiring about 850MB of disk space. By adding a graphical interface (GNOME or KDE), you should expect to use another 1GB of disk space. I usually assume that I will need at least 3GB of disk space for Red Hat Linux, without worrying about running out.
- **Per client space** — You need to decide how much disk space is allowed to each client who will be using your Mini ISP. This is basically to allow the users to add the files needed for their personal Web pages. If, for example, you allow each user to have 500MB of disk space, for five users you would have to add 2.5GB of disk space.

So here are recommendations regarding the amount of disk space you should have available for this project:

- **Recommended Mini ISP** — An 8GB hard drive (or more). With this, you can install all of Red Hat Linux, plus allow space for four to five users who can have 500MB of hard-disk space each.
- **Minimum Mini ISP** — At least a 2GB hard drive. You can do a Red Hat Linux Server installation, then have about 500MB of total disk space left over that your clients can fight for. This might work fine for a Mini ISP that is used mostly for dial-up access but will still allow each user enough space for a bunch of plain HTML Web pages. (You'll run out of space fast if you begin sharing things like video or software packages.)

CD, DVD, or other backup medium

You need some way to install the software and backup the data on your Mini ISP server. Because you will probably use a CD drive to install Red Hat Linux and the Linux Toys packages you need, you might want to use a writable CD drive for all your installation or backup needs. Here are recommendations for backup media you might consider:

- **CD or DVD** — If you already have writable CD or DVD burners on your computer, you can simply use those devices for backing up your clients' data. CD writers and disks are cheaper, but only hold up to 700MB of data each. DVD writers and disks are more expensive, but can hold up to 4.7GB of data.
- **Removable hard disk** — Some of our friends who support large ISP installations use removable hard drives for backing up customer data. Of course, it's more expensive than copying to a CD, but all you have to do is pop in the drive, copy your data over, and remove the drive for safe storage.
- **Network** — If you have another computer accessible to your Mini ISP server over the network, the network itself is a great medium to use to do backups. If the backup host has a large hard disk, you can simply copy the files to that host, using a variety of backup tools in Red Hat Linux (such as `amanda` or `pax`) or by simply copying files across the network (using tools such as `scp`, `cpio`, or `ncftp`).

Modems

You need to decide how many dial-in lines you are going to support and have one modem per line. In our test system, we attach and configure two external modems.

If you are going to buy a modem, steer clear of what are referred to as winmodems. These are not really modems. They are built to be used on Windows systems, and necessary functionality is built into Windows instead of the modems themselves. While some of these modems will work with Linux, you need special drivers. (Refer to `www.linmodems.org` for further information about winmodems that can be made to run in Linux.)

Most external modems (in other words, those that have a separate power supply and plug into a serial COM port on your computer) that are described as supporting hardware UART emulation will work with Linux. You just want to make sure that the modem supports the highest speed that will work on your phone lines (33.6 Kbps is the best you'll get on standard phone lines). Figure 11-2 shows a 56K external data/fax modem that we purchased and tested for this project.

FIGURE 11-2: Most external modems work with Linux (avoid internal winmodems).

If you want to have more modems than you can plug into your two COM ports, you should consider purchasing a serial port board. We didn't test any for this project; however, in the past I have used PC/X boards from Digi International (formerly called Digiboard). Red Hat Linux contains drivers for these boards. If you install the kernel-source package in Red Hat Linux, you can read about how to configure Digi ports boards in `/usr/src/linux-2.4/Documentation/digiboard.txt`.

Network hardware

Your real ISP will tell you what kind of hardware you need to create your high-speed Internet connection. Typically, you will need an Ethernet card on your server to connect to the DSL modem, cable modem, or other equipment.

Chuck's Tech Tip

The network equipment you'll need is a function of the type of data connection you'll have to the Internet. A standard DSL or cable modem connection requires only a basic Ethernet switch. A T1 will require some sort of router. In the old days this was accomplished handily with a Cisco 2500 router. In modern times, a T1 card is pretty much all you'll need to turn a basic PC into a fully functioning router (as well as some expertise with advanced routing protocols).

Step 2: Install Red Hat Linux

Install Red Hat Linux as described in Appendix C. You need to do an Everything installation or a Server installation (adding all server packages). In addition, you may want to add either the GNOME or KDE desktops if you are not comfortable working from the command line.

There are a few special things you can do during installation:

- **Partitioning** — Create a separate `/home` partition that contains enough space for all of your users. For example, by adding a 1GB `/home` partition, you can add 10 users that are each allowed 100MB of disk space.

  Note: Using a feature called disk quotas, you can define how much space each user can consume on a selected partition. To set up disk quotas, you need to install the quota package. Chapter 12 describes how to set up disk quotas.

- **Networking** — If you are using an Ethernet card to connect to your ISP, you can configure that Ethernet card during Red Hat Linux installation. Refer to the "Configure the Internet Connection" section later in this chapter for information on how to configure networking.

Step 3: Install Linux Toys software

There are no Linux Toys packages for this project. Everything is built into Red Hat Linux. (I added this step here so you didn't think you were missing anything.)

Step 4: Configure the network

There are essentially two types of network connections that you have to configure in this project: an Ethernet interface to your ISP and dial-in modems to be used by your clients. Figure 11-3 illustrates these network connections.

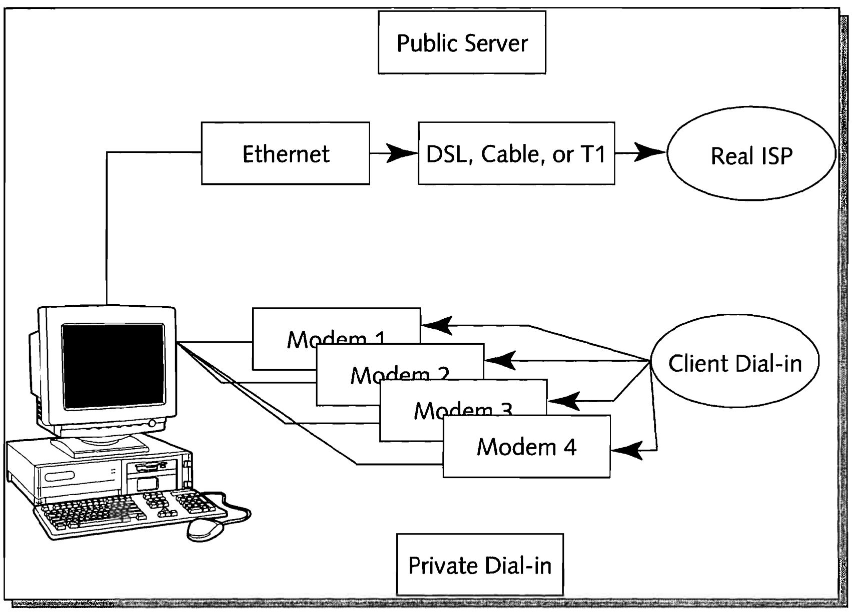


Figure 11-3: Configure connections to your ISP (Ethernet) and from your clients (modem).

Get Internet service, IP address, domain name, and phone numbers

You may need to contact several different places to get what you need to set up your network. Here are some ways to go about getting the IP addresses, domain name, and phone numbers you need:

- **Internet service** — You need to choose an Internet Service Provider (ISP) to which you will connect your Mini ISP. It's up to you to weigh connection speed against cost when you choose the type of service to get. However, you will almost surely want an always-up connection that you can get with DSL or similar service.

  One thing to look out for when choosing an ISP is to carefully read the ISP's terms of use. Some ISPs might explicitly restrict you from sharing your Internet connection.

- **IP addresses** — You probably want to get a permanent IP address that you can assign to the Ethernet network interface. Your ISP will sell you an IP address, usually for a few dollars a month. The reason for a permanent IP address is so you can configure the domain name system (DNS) to find your server when someone tries to contact your server by its host/domain name.

Note

There is a service called Dynamic DNS, which lets you have a domain name assigned to your server without having a permanent IP address. The issue here is that each time you reconnect your DSL service to the Internet, you may be assigned a different IP address.

With Dynamic DNS, any change of your server's IP address is forwarded to the Dynamic DNS service. That service then updates the public DNS records to reflect your new address. While I'm not recommending any of these services (I don't know how well they work, because I don't use them), you can read about these services at `www.zoneedit.com` and `www.ods.org/support.php`.

- **Domain name** — You don't need your own domain name to be a dial-up server. However, for this project we want to offer Web services to our Mini ISP clients. So having our own domain name makes that cooler. In the old days, everyone went to Network Solutions (`www.netsol.com`) to get domain names. Now there are more than 100 registrars. To find a list of accredited registrars, visit the ICANN-Accredited Registrars site (`http://www.icann.org/registrars/accredited-list.html`).

  As part of the registration process for your domain, you need to tell the registrar the IP address of the primary and secondary DNS servers that identify your domain. For our example, we are using the same computer to be the DNS, Web, FTP, and mail server for the domain. So use the one IP address you were assigned by your ISP as the location of the primary DNS server. If you have a friend who has a DNS server, you can ask that person to serve as your domain's secondary DNS server.

- **Telephone numbers** — Contact your local telephone company to have one or more phone lines installed to receive incoming calls to your modems. This is where some of the expense of the project will come in. In our example, we have two modems, so we had two phone lines installed. You can ask your phone company if they can let you use the same phone number to come in on multiple physical lines. That way, your clients don't need to know multiple phone numbers to dial in.

Configure the Internet connection

Because the exact way you configure your connection to the Internet depends on the type of equipment your ISP requires you to use, you have to do a bit of guessing here. Here is a procedure for configuring your connection to the Internet from an Ethernet interface:

1. Start up Red Hat Linux on your computer and log in.
2. From the Red Hat menu on the desktop, click System Settings ➔ Network. You'll see the Network Configuration window and an eth0 interface that represents your computer's Ethernet card.
3. Click the eth0 interface (assuming that the first Ethernet card will be used to connect to your ISP), and click Edit. Figure 11-4 shows an example of the Ethernet Device window that appears.

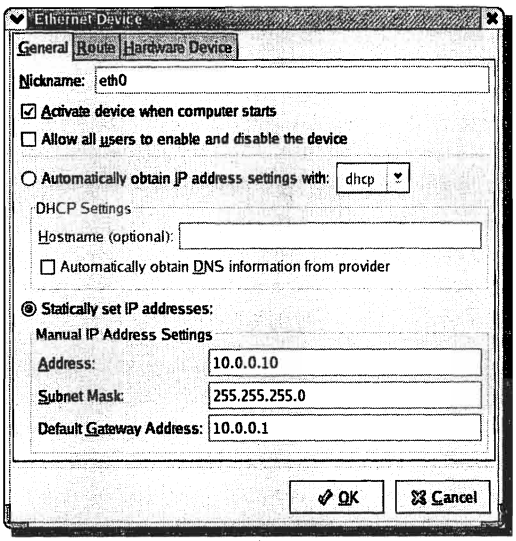


FIGURE 11-4: Define the Ethernet connection to your ISP.

4. There are a few different ways you may need to enter your IP addresses, depending on whether your DSL modem or other device is doing routing or bridging:
   - **Routing** — It may be that the DSL modem or other device is set to your permanent IP address and then to your server with a private IP address. In other words, the outside world may see your DSL modem as address 123.45.67.1, while from your server you see the device as 192.168.0.1 (and assign it as the default gateway). The example in Figure 11-4 shows just such an address arrangement. Your server's IP address is 192.168.0.2 and the DSL modem must be configured to do port forwarding (so requests for Web, FTP, and SSH services can get through).

- **Bridging** — If the DSL modem or other device does bridging, the address you enter manually is the permanent public IP address assigned to your server. It would be as though your server and the ISP were located on the same LAN. Requests for your IP address from the Internet would simply pass through the DSL modem.

   Click OK to accept your changes and continue.

5. Click the DNS tab and add the following information:
   - **Hostname** — Type the fully qualified host name of this server. For example, in the domain `linuxtoys.net`, you might name a host `jukebox`. So you would enter the host name as `jukebox.linuxtoys.net`.
   - **Primary DNS** — Use this server's IP address as the Primary DNS server. That's because you are going to configure this server later as a DNS server.
   - **DNS Search Path** — Type your domain name here (for example, `linuxtoys .net`). This just tells your computer that if someone indicates a host name (with no domain name at the end), it should look in the current domain to find that host name.

   Click OK to accept your changes and continue.

6. From the main Network Configuration window again, click Deactivate; then click Activate to have the changes you made take effect.

At this point, you should be able to communicate with your ISP. Using an IP address you know belongs to a computer at your ISP or on the Internet, type the following from a Terminal window (replacing `123.45.67.1` with the real IP address):

```
$ ping 123.45.67.1
```

Configure the dial-in connections

In Linux, dial-in connections are done by attaching a modem to a serial (COM) port on your computer (or using an internal modem), plugging the modem into a phone line, and configuring Linux to listen for calls on that modem. As noted in the "Gather Computer Hardware" section, the first trick is to get a real modem.

For this example, you are going to attach one external modem to each of your COM1 and COM2 ports on your server computer. In Linux terms, the connections to those ports are represented by special files in the file system: `/dev/ttyS0` and `/dev/ttyS1`.

To begin, connect your first modem to your COM1 port. Use a standard phone cable (RJ-11) to connect your modem to the phone line that you dedicate to dial-in service. Then run the following procedure (as root user from a Terminal window):

1. Edit the `/etc/inittab` file and add the following lines:

```
S0:2345:respawn:/sbin/mgetty ttyS0
S1:2345:respawn:/sbin/mgetty ttyS1
```

These lines start `mgetty` processes (when your system boots) that listen for connection requests on your COM1 (`ttyS0`) and COM2 (`ttyS1`) ports. If you have modems on COM3 or COM4 ports, you should create additional lines (for `ttyS2` and `ttyS3`). If you have a multi-ports board, refer to the documentation for that board to determine which devices to listen on.

2. Add the following line to the `/etc/mgetty+sendfax/login.config` file (you can start by uncommenting the AutoPPP line that is already in the `login.conf` file):

```
/AutoPPP/ - a_ppp /usr/sbin/pppd auth \
-chap +pap login debug modem \
crtscts proxyarp lock ms-dns localhost
```

The previous information should all be on one line (I added the backslashes to indicate that each subsequent line should actually be joined together). This line causes the pppd daemon to be launched in response to connections on the mgetty line that receives an incoming phone call. That daemon sets up a Point-to-Point Protocol (PPP) connection between the computer dialing in and the server. Here is what the options do:

- **auth -chap +pap login** — Causes PAP style of authentication to be used, but not CHAP. PAP is the authentication type that Windows clients will expect and that Linux clients can do as well. Login information that the client enters will be authenticated using PAP.
- **debug** — Logs the content of control packets. By default, that information will be copied to the `/var/log/messages` file. If the dial-in doesn't work, look in that file to find the problem.
- **modem** — Tells the PPP daemon to wait for Carrier Detect (CD) before opening the device that connects to the modem. This is the right way to act with modems, since you don't prompt for a login until a call comes in.
- **crtscts** — Indicates that the modem should use hardware flow control
- **proxyarp** — Allows PPP to forward packets from the server to the Internet
- **lock** — Causes the PPP daemon to lock the device, so no other processes can connect to the modem while it is in use
- **ms-dsn localhost** — For Microsoft Windows clients, this option can pass the name of the DNS server so the client doesn't have to enter it manually. In this case, since our DNS server is on the same dial-in host, we have localhost as the location of the DNS server. (You could use an IP address instead of localhost.)

3. Add a line to the `/etc/ppp/pap-secrets` file for each user that you want to allow to dial in. Here are a couple of examples:

```
jwsmith    *    myp9d3wd      10.0.1.10
mcjones    *    4yerp2xwRd    10.0.1.11
```

In these examples, the users logins jwsmith and mcjones can connect from any server (*) and log in using myp9d3wd and 4yerp2xwRd as passwords, respectively. The IP addresses assigned to the PPP sessions started by those accounts are 10.0.1.10 and 10.0.1.11, respectively.

4. You need to start the mgetty processes you configured in the first step. Type the following command to reread the inittab file and start mgetty processes:

```
# telinit q
```

The mgetty processes should start running and the users you configured should be able to log in to your server at this point. However, for the users to be able to use the Internet through that connection, you need to configure the server to allow packet forwarding and masquerading (or NAT) to get through the firewall.

Allow packet forwarding and masquerading

There are essentially two steps for allowing packet forwarding and routing between your dial-up users and the Internet. Those steps are described as follows:

Note Whole books are devoted to firewalls and routing. I tried to provide a couple of steps that will simply do what we need to let dial-up users route through our server to the Internet. If you vary much from our canned demonstration, you will need to find other resources for details on how firewalls and routing work in Linux. *Red Hat Linux Firewalls* by Bill McCarty (Red Hat Press/Wiley, 2003) expands on my firewall coverage in this book and in *Red Hat Linux Bible*.

1. To allow packet forwarding in the Linux kernel, add the following line to the `/etc/sysctl.conf` file:

```
net.ipv4.ip_forward = 1
```

2. Assuming that you configured an iptables firewall when you installed Red Hat Linux (which you do by default), you need to tell the firewall to masquerade the addresses that came in through the dial-up PPP connection.

   If you used the private IP addresses described earlier in the pap-secrets file (addresses 10.0.1.10 and 10.0.1.11), this procedure should work fine.

```
# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
# iptables -A FORWARD -s 10.0.1.0/24 -j ACCEPT
# iptables -A FORWARD -d 10.0.1.0/24 -j ACCEPT
# iptables -A FORWARD -s ! 10.0.1.0/24 -j DROP
# iptables-save > /etc/sysconfig/iptables
```

   Essentially, we ran commands that tell the kernel that it is okay to masquerade addresses to the eth0 interface (presumably, the Ethernet card that gets you to the Internet connection). Any packets received by the firewall where the source (`-s`) or the destination (`-d`) are on network 10.0.1 should be accepted and forwarded. Packets not on that network should not be forwarded. Assuming that the current running firewall has a sane set of rules, you save those rules permanently (so they are re-read each time the firewall is restarted) by running the preceding `iptables-save` command.

Step 5: Configure the domain

You should now have a connection to the Internet with a permanent IP address assigned to your server. You should also have a registered domain name that points to this IP address. Next, you need to configure DNS.

The main job of a Domain Name System (DNS) server is to take requests for host names of the domain it serves and to translate them into IP addresses. For example, if your domain were `example.com`, someone browsing the Web trying to get to visit `www.example.com` will be directed to the IP address provided by your DNS server. The DNS server can do the opposite as well (return a host name when a client asks who owns an IP address).

Our example of a DNS server (we'll call it `example.com`) provides service for:

- A single domain
- A single host (DNS, Web, mail, and FTP service are all on the same computer)
- An additional secondary server (find a friend to be your DNS secondary)

You need to configure an `/etc/named.conf` file and then create a zone file in the `/var/named` directory. Figure 11-5 shows two files: `/etc/named.conf` (on the left) and `/var/named/db.example.com` (on the right).

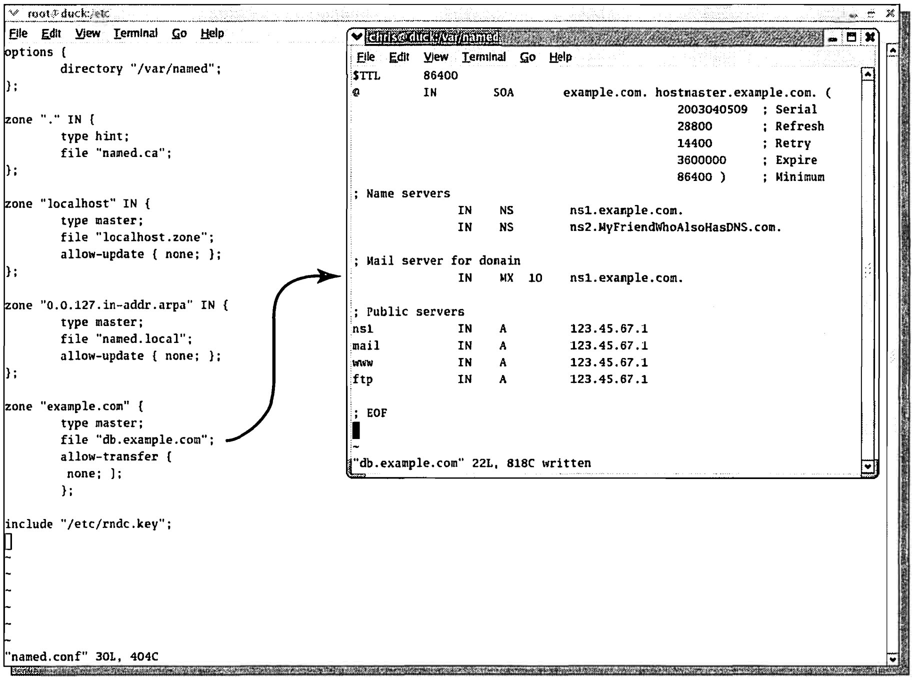


FIGURE 11-5: Create a named.conf file, then one or more zone files.

The named.conf file sets options for the name server (in this case, simply the location of the zone directory). It then defines the zones associated with the name server you are creating. The first three zones shown in the named.conf file in Figure 11-5 (named.ca, localhost.zone, and named.local) already exist. The last file (db.example.com) we supplied. Here is how you can modify the named.conf file and create a new file for your domain in the /var/named directory:

1. Open the /etc/named.conf file in any text editor as root user from a Terminal window.
2. Create an entry for your own domain in the /etc/named.conf file (similar to the one shown in Figure 11-5). In the following example, change example.com to your domain name:

```
zone "example.com" {
type master;
file "db.example.com";
allow-transfer {
none; };
};
```

3. Save the change and close the /etc/named.conf file.
4. Change to the /var/named directory:

```
# cd /var/named
```

5. Create a file using the name you just added to the named.conf file (for example, db.example.com) by opening that filename in any text editor. Here is a /var/named/db.example.com file you could start with:

```
$TTL      86400
@         IN        SOA       example.com.
hostmaster.example.com. (
                                      2003040509   ; Serial
                                      28800        ; Refresh
                                      14400        ; Retry
                                      3600000      ; Expire
                                      86400 )      ; Minimum
; Name servers
               IN    NS        ns1.example.com.
               IN    NS
ns2.MyFriendWhoAlsoHasDNS.com.

; Mail server for domain
               IN    MX  10    ns1.example.com.

; Public servers
ns1            IN    A         123.45.67.1
mail           IN    A         123.45.67.1
www            IN    A         123.45.67.1
ftp            IN    A         123.45.67.1

; EOF
```

Now modify that file as follows (making sure to get all the dots and semicolons right along the way):

- Change all instances of `example.com` to the name of your domain. For example, for the `quantumlinux.com` domain, you would have `quantumlinux.com.`, `hostmaster.quantumlinux.com.`, `ns1.quantumlinux.com.`, and so on. (Be sure to leave the ending dot on those names.)
- Change `ns2.MyFriendWhoAlsoHasDNS.com` to the name of the host computer that has `agreed` to be a secondary DNS server for your domain. (Your registrar may require a secondary DNS server. If you don't have one and it's not required, simply delete this line.)
- Change all `instances` of 123.45.67.1 to the real IP address of your server.

6. Save the change and close the `/var/named/db.example.com` file (actually, the name you changed that file to).
7. If you are ready to start your DNS server, type the following lines as root user from a Terminal window:

```
# service named on
# chkconfig named start
```

Your DNS server should now be up and running. Here are a few things you should know about the server you just created:

- Each time you make a change to the DNS server, you need to edit the `named.conf` file to increase the serial number and then restart the `named` service. Often, administrators use the current date and then increment a two-digit number at the end. For example, May 7, 2003, would appear as 2003050701. If you change DNS again on that day, you could change it to 2003050702. (To restart DNS, as root user type **`chkconfig named restart`**.)
- Anyone on the Internet will be able to reach your server using a variety of names. For example, for the `quantumlinux.com` domain, people could reach the server by the names `ns1.quantumlinux.com`, `mail.quantumlinux.com`, `www.quantumlinux.com`, and `ftp.quantumlinux.com`.
- The MX record identifies the location of the mail server for the domain. For example, in the `quantumlinux.com` domain, `ns1.quantumlinux.com` would be the server to receive mail for `chris@quantumlinux.com`.

Step 6: Configure the Web server

Use the descriptions in Chapter 14 to configure your Web server. In addition to that, to allow users to configure their personal Web pages, change the `UserDir disable` line to read:

```
userDir public_html
```

This will allow each user to place personal Web pages in the `public_html` directory in their home directory. Next, restart the Web server:

```
# service httpd restart
```

Step 7: Configure the FTP server

Use the descriptions in Chapter 12 to configure your vsFTPd FTP server. In addition to that, add the following lines to the `/etc/vsftpd/vsftpd.conf` file:

```
anonymous_enable=No                           # Only real users get in
chroot_list_enable=YES                        # Users stay in home dir.
chroot_list_file=/etc/vsftpd.chroot_list      # List of exceptions
```

These lines make it so that all users, when they connect to the server using FTP, cannot access (or even see) anything outside of their own home directory. If you want a user to be able to traverse the whole computer on an FTP connection, add the user's name to the `/etc/vsftpd.chroot_list` file. No `anonymous` users can log in from FTP.

After you change the vsftpd.conf file, restart the FTP server as follows:

```
# service vsftpd restart
```

Step 8: Configure the mail server

We chose postfix as the mail transfer agent because it is easier to configure and many consider it to be more secure than sendmail (the other mail server software that comes with Red Hat Linux).

Start Up postfix

To create a simple configuration for your postfix mail server, edit the `/etc/postfix/main.cf` file (as root user from a Terminal window). Then find the `myhostname` and `mydomain` lines and change them to include the name of your mail server and domain, respectively. For example, following the preceding example where we created a domain named `quantumlinux.com`, the entries would read as follows:

```
myhostname = mail.quantumlinux.com
mydomain = quantumlinux.com
```

To start postfix, type the following as root user from a Terminal window:

```
# service postfix on
# chkconfig postfix start
```

Start Up ipop

The ipop3d daemon can be used to listen for requests from mail clients to get their mail from the server. To configure the ipop3d mail client, type the following (as root user from a Terminal window):

```
# chkconfig ipop3 on
# service xinetd restart
```

At this point, anyone who asks to connect to your server requesting Post Office Protocol service to get their e-mail will be able to get to their mailbox (as long as they provide a valid user name and password).

Step 9: Add user accounts

You need to create a user account for every person you want to be able to log in and use your server. That same login will be used for remote login (via SSH), Internet access, mail server access, and most other services you may care to add.

Here are the steps you can go through to create user accounts (as root user from a Terminal window):

1. Add a `public_html` directory to `/etc/skel` so that directory will be created in each user's home directory when you create each user. Also change the permissions so the Web server can read from it. Type:

```
# mkdir /etc/skel/public_html
# chmod 755 /etc/skel/public_html
```

2. Add each user account. For example, to add a user named mcwatson, you could type the following:

```
# useradd -m mcwatson
```

3. Add an initial password for the user. For example, as root, type:

```
# passwd mcwatson
```

4. Change the permission of the new home directory so that execute permission is open (otherwise, the Web server won't be able to read the Web content):

```
# chmod 711 /home/mcwatson
```

Users can change the password later by logging in and running the `passwd` command again.

Step 10: Open the firewall

Ways to configure firewalls are described in Chapter 7. Once your firewall is up and running, you need to open up some holes in it to allow network services to be used by people using the server from the Internet. Here is a list of ports that need to be opened to allow the services described in this chapter:

- **Web server (http)** — Port 80
- **FTP server (ftp)** — Ports 20 and 21
- **DNS server (domain)** — Port 53
- **Remote login (ssh)** — Port 22
- **Mail service (smtp)** — Port 25
- **Mail retrieval (pop3)** — Port 110

After you have updated your firewall rules, restart the iptables firewall as follows:

```
# service iptables restart
```

That completes the basic configuration of your Mini ISP server. The next section describes how to support your users and maintain your Mini ISP.

Note

If people cannot access services from your server, try flushing the firewall rules temporarily by typing `iptables -F`. If the service is then available, restart iptables and step through the process of finding which port needs to be open. (Flushing your firewall presents a security risk so, if possible, try to test the firewall in a safe environment.)

Supporting Users and Maintaining the Mini ISP

While my friend Pat was with me taking the fine penguin images you see in this book, his brother called him. He had given his brother an e-mail account on his server, and his brother was having trouble getting through to the server. After about 10 frustrated minutes of "the server's fine, tell me what you did," Pat hung up and went back to the penguins. If you set up a Mini ISP, expect such calls.

Whether or not you provide the 24/7 support, if you give a friend a user account, they're going to expect support. To make that job easier, there are a few things you might want to do on occasion to keep your Mini ISP working well. This section describes some support and maintenance issues.

Prepare user account information

You should prepare a sheet of information for each client that can be used to connect to your service. Here's the information you need to provide to each client, with examples of that information:

Information	*Example*
Phone number	201-555-1212
User name	mcwatson
Password	MkMeyeDay
Mail server	`mail.example.com`
FTP server	`ftp.example.com`
Personal Web pages	`www.example.com/~mcwatson`

Be sure to use good passwords. Also, of course, you will use your own domain and host names for the mail and FTP servers.

Perform administrative tasks

Having a public server is like owning a store in a major city. You have to expect that someone is going to be testing whether he or she can sneak in the back door or even come in the front door and try to shoplift something.

Even though you may not have critical data on your server, it would still be annoying if someone were to break in and trash your computer. Or, possibly worse, they might just take over your machine and use it to forward SPAM or launch attacks on others. Here are a few tasks you can do to keep your server safe and up and running:

- **Backups** — Make a schedule for backing up your server's data. With the way we have described this project, you want to back up at least the `/home` and `/etc` directory structures. The frequency of your backups depends on how critical your data resources are and how much they change. Think of it this way: Could you recover from a total loss of data from today, the past week, or the past month? You should backup whatever you can't afford to lose.
- **Log files** — Linux system log files are contained in the `/var/log` directory. Logs of particular interest include `messages`, `secure`, `maillog`, and `vsftpd.log`. Also, for your Web site, in the `/var/log/httpd` directory, you might be interested in the `access_log` and `error_log` files.

Chuck's Tech Tip

Back in the old days, you'd run a full backup once a week and then take incremental backups on a daily basis, all done on magnetic tape. It is still important to take regular backups, but for most situations, there's little reason anymore to do this on magnetic tape. Hard drives are far cheaper than tape drives and can be written over many more times than a magnetic tape could. Consider an investment in "hot swappable" hard-drive bays to run your backups to. Do a full backup each night to the removable drive and bring it to a secured location during the day. The primary benefit is that recovering data from a hard drive is much faster than it is from a tape drive. Another backup scenario we use quite often at Quantum Linux Laboratories is rsync'd LVM snapshots. An elaborate script creates a new LVM snapshot volume for each of our customers and the latest data is rsync'd to the master volume. This enables us to do a full backup of a company's data in minutes and keep all revisions of that data in a minimal amount of space.

Using Your Mini ISP

Armed with the information in the paper you just gave them, your clients should be able to set up their own dial-up configuration to reach your server. Once they're on, there are lots of ways they can use it. Here's a rundown:

- **Configuring dial-up networking** — On most Windows systems, that consists of opening the Dial-up Networking window (from My Computer) and adding a new connection. That procedure prompts for the phone number, user name, and password provided on the paper you filled out.

- **Upload Web pages** — Using any ftp client program, users can use the same account name and password to copy Web pages to the `public_html` directory in their home directory.
- **View Web pages** — For a user named `jwsmith` who copied a Web page named `info.html` to their his `public_html` directory in the domain `example.com`, the URL to that page would be as follows:

  ```
  http://www.example.com/~jwsmith/info.html
  ```
- **Access e-mail** — When the user account was created, so was an e-mail account of the same name. Users can fire up their favorite e-mail reader and create an account using the e-mail address and password you gave them. They also need to know that it is a POP server that is not doing encryption.

Summary

While a *real* ISP requires hardened servers and constantly diligent support, many of the features that a professional ISP uses are built into Red Hat Linux, as well as other Linux distributions. This Linux Toys project is meant to describe how to set up the basic services needed to build a Mini ISP. The result is a single computer that lets clients dial-in, browse the Internet, access personal e-mail accounts, publish personal Web pages, and upload files using FTP. For more serious server development, you can refer to *Red Hat Linux Bible* or one of many specialized books on each of the subjects covered here.

Be a Web-Hosting Service

chapter 12

in this chapter

☑ Preparing your Web server

☑ Serving multiple domains on a Web server

☑ Adding users and FTP, mail, and other services

☑ Monitoring your Web server

If you have a PC, Red Hat Linux, and an always-up connection to the Internet, you have all the tools you need to publish content on the Web. Get yourself a public IP address (so the Internet always knows where your server is), and have your friends and family each buy their own domain names, and you could be a Web-hosting service!

Like the Mini ISP project described in Chapter 11, this chapter steps you through the hardware and software issues related to setting up an Internet server. The tools described here are the same ones used by many professional Web-hosting providers. In fact, the Apache Web server software we describe here is used worldwide on nearly every popular computing platform (including Linux, of course).

Step through the procedures here and you will end up with a simple configuration that offers all the basic features you would expect from a professional Web-hosting service. Backing up your Web service is also a full complement of tools for monitoring and maintaining the server. (It is running in Linux, after all!)

Here are some of the things you could do with this Web-hosting service:

- Create a Web site for a local charity event.
- Put your resume online to help in your job search.
- Publish family pictures to share with others around the world.
- Display the rates for your son's lawn-mowing business.

For a few dollars a year, each of these sites could even have its own domain name. You could get: `bakesaleformycharity.org`, `chrisplease givemeajob.com`, `thejonesfamilyonsafari.net`, or `readyto mowcheap.com`. Then start printing business cards and flyers with your very own Internet address on them.

Even though we're calling this a toy, the tools we describe here are exactly the same ones used every day by real Web-hosting services. So you could run a business-oriented Web site that sells products, processes credit cards, and tracks inventory and the like by using Red Hat Linux as the platform.

But, for goodness sake, you did after all buy *Linux Toys*. So we're going to assume that you don't have the level of commitment for a massive

computing infrastructure and 24/7 support. This project should, however, work well for your bake-sale Web site. And maybe some day you can buy a bigger book on Web hosting and start hosting international businesses (see Figure 12-1).

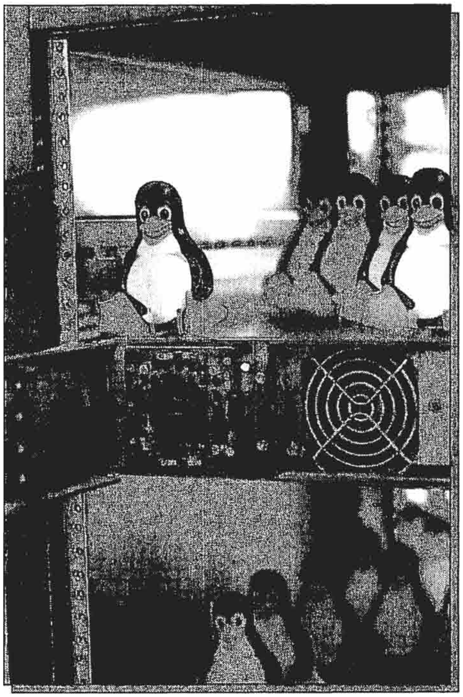


FIGURE 12-1: Host Web sites for your friends, family, and unimportant international businesses.

> **Note** In the old days, one host computer contained one Web site. These days, a feature in the Apache Web server allows us to do *virtual hosting*. Virtual name-based hosting lets many different Web sites (such as `linuxtoys.net`, `example.net`, and so on) reside on the same physical computer. Based on the site someone requested from a Web browser, the Web server knows to serve up different content.

Planning Your Web-Hosting Service

Does Web hosting mean just letting your clients publish a few plain HTML Web pages on the Internet? Well, it can . . . but why should it? There are tons of features that are free with Red Hat Linux that a lot of Web-hosting services are charging big bucks for.

Stepping through the next set of questions will help you plan how to set up your Web server. It will also help you write your own version of "Chuck and Chris's Web Hosting" brochure (to hand out to your family and friends).

What comes with each Web site?

Part of our exercise is to set up virtual hosting (in other words, Web sites for multiple domains on one physical computer). So our first assumption is that all of your customers will want their own domain names. Along with a Web site for their domain names, your customers will get these basic services:

- **Web site** — If your client's domain is `example.com`, anyone on the Internet can reach that domain's Web site at `www.example.com` from any Web browser. Clients add their own Web content (limited only by the disk space and services you allow them to have).
- **E-mail accounts** — You can create an unlimited number of e-mail accounts on your server. So multiple users for each domain could have their own addresses, such as `chrisn@example.com` or `chuckw@example.com`.

  You can set up mail accounts to be accessible via Post Office Protocol (POP), so clients can read e-mail from their own mail clients. For our configuration, we're going to require that each e-mail account be unique on the Web server. In other words, e-mail to `chrisn@example.com` or `chrisn@linuxtoys.net` on the same server would both go to the same mailbox.
- **FTP access** — To place their Web content on the server, you grant you clients FTP access to the Web server.

What resources can each Web site use?

The size of the computer's hard disk and bandwidth to the Internet place limits on what your Web server can do. You can set boundaries that limit the amount of storage space and the amount of traffic that each client can consume. (If they want more, they're going to have to pay for it . . . or at least ask nicely.)

Disk storage

A few plain HTML Web pages aren't going to take up a lot of disk space. But if someone is going to be distributing audio, video, or databases of information, you need to consider limiting how much disk space they can consume.

Introductory packages at a Web-hosting service may allow 100–500MB of disk space to customers. To give you an idea of what this means, here's a rough estimate of what a client could fit in 100MB of disk space:

- 10,000 simple HTML Web pages (averaging about 10 kilobytes each) or
- 2,000 JPEG images (averaging about 50 kilobytes each) or
- Two whole music CDs, compressed with Ogg Vorbis (50MB each) or
- One video clip containing 1–3 minutes of content, compressed with MPEG-4

To enforce control over how much each client can consume, you can use a feature called disk quotas that comes with Red Hat Linux. (I describe how to use disk quotas later in this chapter.)

Data transfer

Many inexpensive hosting services offer about 10–30GB of file transfer per month before they begin to charge extra to their customers. You can monitor the amount of traffic that is coming to each client you host by the number of hits, the amount of data transferred, or the number of pages viewed.

Tools such as Webalizer, which comes with Red Hat Linux, can be set up to track the activity on the Web sites you serve.

Chuck's Tech Tip

If you plan to offer services in addition to Web hosting, such as mail and FTP, you'll have to take into the account the data transfer taken up by those services when calculating a customer's total data transfer. The best way to do that is to watch the log files for each service and add that to the total given to you by your Web statistics package (such as Webalizer). It is much too labor intensive to parse through log files manually. For that, you should find or write a script to do the work for you.

What content types will you allow?

An inexpensive PC can serve plain HTML pages all day without chewing up much of your computer's resources. Allowing your clients to provide other types of content, however, can be a big drain on resources.

Of course, allowing different types of content can also make a Web site more dynamic and interesting. Here are some types of content you might consider allowing clients to include on your Web server:

Note

This is just a sampling of content you can include on a Web page. Any type of content that can be held in a file can be downloaded from a Web site or FTP site. Likewise, many different programming languages can be used to write code that is incorporated into Web content.

- **CGI scripting** — Allowing CGI programs provides a way for your clients to include more interactive content on their Web pages. CGI scripts are often used to implement forms on a Web page and access databases. The script itself can incorporate different programming and scripting languages, such as Perl, Python, and TCL.
- **PHP** — A scripting language that you can embed in HTML code. Clients can use this instead of CGI scripting to add dynamic content. You can enable PHP on your Apache Web server to allow clients to include PHP code in their HTML pages.
- **Perl** — A popular scripting language used to create CGI scripts for Web pages. There are many different Perl software libraries available with Red Hat Linux for doing different types of jobs.
- **Database** — Red Hat Linux contains both MySQL and PostgreSQL database client and server software. If you allow it, your Web-hosting clients can both create and access databases from their Web sites.
- **Secure server access** — By adding secure server access (SSL) support, you can improve the security of communications between your Web server and the clients who browse the site.

What should your service provide?

If you are just serving your Aunt Millie's recipe for strawberry jelly (or for your Aunt Pearl's jam), you may not feel the need to put a huge amount of time and money into your service and support. If the data you are serving is critical, however, you should consider how much money, time, and effort you want to put into your Web server.

- **Data backups** — If the Web sites you are serving contain any type of commerce, you need to back up the data on your Web server on a basis that is beyond anything we cover here.

Cross-Reference

See Chapter 11 for a discussion of different backup methods that may interest you.

- **Log monitoring** — The activities of your Web server and FTP server are logged in files in the `/var/log` directory. You can monitor those files manually or use various tools (such as `logwatch`) to scan log files and forward messages that contain potentially dangerous activities to your mailbox.
- **Firewalls** — Every computer connected to the Internet, but especially servers, should have a firewall between itself and the Internet. For a server, the most important feature of the firewall is to define which types of services someone can request from the server. (All other requests should be denied.)

  Typically, the firewall machine is separate from the server. For our low-cost Web-hosting service, however, we're going to have the firewall configured right on the Web server itself.
- **Spam killers** — There are lots of tools around today to deal with spam (unsolicited, mass-distributed e-mail). Red Hat Linux comes with spamassassin, which uses a set of rules to guess at whether a particular message is spam or not.
- **Web statistics** — So that your clients can see how many visits they get, you can gather and provide statistics on each Web site's activities. There are tools in Red Hat Linux that let you produce graphical representations of traffic to a site, as well as raw data that a client can use with a variety of tools.

What do you need from a service provider?

The same considerations for being a Mini ISP (see Chapter 11) apply to choosing an ISP for your Web-hosting service. You need bandwidth to the Internet, a public IP address, and an ISP that allows you to run a Web server (some ISPs consider that a premium service).

One additional consideration: In a pure "Web-Hosting" situation, you might consider co-locating your computer at the site of the service provider. It may cost a few dollars more per month, but you will probably get much better bandwidth and possibly more reliable service. They usually have someone on site that can reboot if necessary and some sort of universal power supply (UPS) to keep running in case the power goes out.

What do you need to tell your clients?

Just because it is easy to do illegal activities on the Web doesn't mean that you won't get yourself thrown in jail for it. If you own the computer and are hosting the service, law enforcement might not care that your Uncle Sidney was just up to his old tricks again.

You should consider putting together your own "Terms of Use" document that describes what someone can and can't do with your hosting service. Here are some DON'Ts you might want to include:

- **Don't store or transmit any data or files that violate local or federal laws.** This includes any copyrighted data or software.
- **Don't participate in any abusive behavior.** This includes hacking, mail bombs, flames, or spam.
- **Don't transmit or sell pornography.**
- **Don't do anything to harm the security of this server.**

Then be sure to tell your Web-hosting clients that they are responsible for their own actions. To protect yourself, it doesn't hurt to keep an eye on the content you are hosting. Watch for big spikes in activity as a red flag that something might be up.

Chuck's Tech Tip

As a general rule, service providers are not responsible for the content their customers make available. However, flawed legislation like the DMCA (Digital Millenium Copyright Act) has reverted some of that burden to the service providers. This is loosely akin to forcing the post office to open every letter and package that it handles to ensure that it is in compliance with federal, state, and local laws. Under the DMCA, service providers are given a safe harbor wherein they have a limited amount of time to act when informed of a possible legal violation. Inaction will result in criminal and civil penalties. This is usually enough to scare small-time providers with limited budgets into compliance. Whether the original claim is justified or not is irrelevant.

Configuring Web Hosting

Once you install Red Hat Linux, connect it to the Internet, and configure the Apache Web server software, your Web server is essentially up and running. The rest of the configuration you choose to do depends on:

- How you want to let your clients load their Web content
- What additional services you want your clients to have
- How much security and support you want to give your clients

The setup you use with this project is similar to the Mini ISP (Chapter 11). So, you can start with the first few steps in that procedure, with some slight modifications, before continuing with the steps in this project.

Step 1: Set up the server

Start by running the first five steps in Chapter 11. Here is how you should modify each of those steps to work with this project:

- **Step 1 (Hardware)** — You can follow the same hardware requirement as described in this step. One exception is that you don't particularly need a modem (unless your connection to the Internet is through dial-out, which I don't recommend for this project).
- **Step 2 (Red Hat Linux)** — You can follow the same installation suggestions as in Chapter 11. An Everything or Server install type of Red Hat Linux should work fine. If you want to be able to restrict how much disk space each virtual host can use, be sure to create a separate `/home` partition as noted.
- **Step 3 (Linux Toys)** — As with the Mini ISP, there is no specific Linux Toys package for this project.
- **Step 4 (Network)** — You can follow this procedure, with the exception that you don't have to configure a modem.
- **Step 5 (Domain)** — You can configure DNS for each virtual Web host that you add to your system.

By this point, you should have the hardware and software set up for your Web server. You should also have a high-speed Internet connection from a public IP address on your computer. DNS records should exist for each of the virtual hosts you are about to configure.

Step 2: Add user accounts

You need to create a user account for each virtual Web server running on your computer. That user will have permission to upload files to your Web server (via FTP) so that the files immediately become part of that domain's Web site.

All user names on the same computer must be unique, regardless of which domain's virtual Web server they are serving. In other words, only one user named `chuckw` could be assigned on the computer. If `example.com` and `example.net` Web servers were both on the same machine, the users `chuckw@example.com` and `chuckw@example.net` would both point to the same user account. That concept would apply to using those account names for e-mail, FTP access, or regular Linux login.

1. Type the following command as root user (from a Terminal window).

   Note

   If you prefer graphical tools instead of commands, you can add a user with the redhat-config-users window. To open that window, click the Red Hat menu button → System Settings → Users and Groups.

   ```
   # mkdir /etc/skel/public_html
   # chmod 755 /etc/skel/public_html
   ```

   This creates a `public_html` directory that is automatically added to each new user's home directory. The `chmod` command opens permission for anyone to read and execute anything contained in the `public_html` directory. (You can put any file or directory in `/etc/skel` that you want every new user to have.)

2. You could add a user account for each domain you host (or have multiple domains maintained by a single user by setting up multiple directories accessible from that user

account). For example, if Chuck is the Webmaster for the `example.com` domain Web server, you can create an account for the user name `chuckw` as follows:

```
# useradd -m chuckw
```

This adds the `chuckw` user account and creates the `/home/chuckw` directory (complete with its own `/home/chuckw/public_html` directory).

3. Add an initial password for the user. For example, to add a password for chuckw, type the following:

```
# passwd chuckw
```

Type the password (then again when prompted).

4. Turn on execute permission on the user's home directory; otherwise the Apache Web server daemon won't be able to get at the Web content. For example, for the user chuckw, you could type:

```
# chmod 711 /home/chuckw
```

5. Optionally, you can restrict the users so that they have FTP access only, but no shell account. This is an added measure of security if the users need only to add and remove files related to the Web server they control. For example, to restrict the account of `chuckw`, open the `/etc/passwrd` file and change `/bin/bash` (or other shell shown) to `/sbin/nologin`. It will appear as follows:

```
chuckw:x:501:501::/home/chuckw:/sbin/nologin
```

In this case, chuckw will be rejected if he tries to log in at a login prompt. However, if he tries to log in to the server with FTP (with a valid password) he will be accepted.

Step 3: Configure the FTP server

Our virtual Web server needs content! To be able to put content on the Web server, we are going to allow the user assigned to each virtual Web server (represented by the domain name) to upload files for their server using any FTP client program. To do that, we are going to configure the vsFTPd FTP service in Red Hat Linux.

Note You should read through the comments in the `/etc/vsftpd/vsftpd.conf` file. This procedure will quick-start your FTP server to do what we need it to. However, your mileage may vary.

1. Using any text editor, from a Terminal window or any shell, open the `/etc/vsftpd/vsftpd.conf` file as root user. Then make the following changes to that file:

```
anonymous_enable=NO                        # Only real users get in
chroot_list_enable=YES                     # Users stay in home dir.
chroot_list_file=/etc/vsftpd.chroot_list   # List of exceptions
```

To get the first line, change the existing `anonymous_enable` line from `YES` to `NO` (as shown). This prevents anonymous users from connecting to your Web server through

FTP. For the next two lines, remove the comment characters (#) in front of each. This causes all users (except those listed in `vsftpd.chroot_list`) to start their FTP sessions for their own home directories.

2. Type the following to create an empty vsftpd.chroot_list file:

```
# touch /etc/vsftpd.chroot_list
```

3. You can start the vsFTPd server; then set it to start immediately. Here's how:

```
# service vsftpd start
# chkconfig vsftpd on
```

Step 4: Configure the mail server

Each user account you add to the computer housing your Web server will automatically have its own e-mail account. You can configure that e-mail account using postfix, as described in the "Configure the mail server" step in Chapter 11.

Note

The postfix mail server has relaying disabled by default. If you want to double check that nobody can use your server to relay spam, go to `http://www.abuse.net/relay.html`. Once you register with abuse.net, you can have it send a series of checks to make sure that you can't be used as a mail relay.

A mail server allowing open relay will forward e-mail messages that are not from or for local users. Spammers use these machines to disguise where the spam messages are coming from. If your mail server is taken over by spammers, you risk having your server blackballed as a spamming server.

Step 5: Configure the Web server

The Apache Web server (httpd package) comes with Red Hat Linux. Based on the latest Netcraft statistics, it is the most popular Web server in the world. We're not going to get into most of Apache's monstrous feature set here (entire books have been written on that). But we do tell you how to configure Apache to have multiple virtual Web servers running on it (which is powerful enough if you're just starting out).

To configure your Web server, at this point in the procedure you should have the following information for each domain that you are ready to host:

- **Domain name** — Each client should have purchased a domain name and identified your computer's IP address as the primary DNS server for that domain.
- **User name** — You should know the user account on your Web server that is responsible for uploading the content for the domain's Web server.

Armed with this information, you are ready to set up your Web-hosting service. After you have logged in to Red Hat Linux as the root user, open the `/etc/httpd/conf/httpd.conf` file with a text editor from a Terminal window (or other shell). Here are some of the lines you should add or change:

1. Change the address shown on the following line (`root@localhost`) to any address you want to receive mail regarding problems with your server:

   ```
   ServerAdmin root@localhost
   ```

2. Turn on the VirtualHost feature by removing the comment character (#) from in front of the following line:

   ```
   NameVirtualHost *
   ```

3. Create a VirtualHost container for each virtual host you are serving on your Web server. Add your virtual host containers to the end of the file. The first VirtualHost you define will be the one returned if someone contacts your server by IP address, instead of by hostname. Here is an example:

   ```
   <VirtualHost *>
      ServerAdmin chuckw@example.com
      DocumentRoot /home/chuckw/public_html
      ServerName www.example.com
      ServerAlias example.com
      ScriptAlias /cgi-bin /home/chuckw/public_html/cgi-bin
      ErrorLog logs/example-error_log
      TransferLog logs/example-access_log common
   </VirtualHost>
   ```

   In this example, we are creating a virtual Web server for the `example.com` domain. Here is what each line in the VirtualHost container means:

   - E-mail regarding the server is directed to `chuckw@example.com`.
   - The `public_html` directory in the user chuckw's home directory is where the content for the domain's Web server is placed.
   - Any Web browser that tries to connect to `www.example.com` (ServerName) or `example.com` (ServerAlias) is directed to the content for this Web server.
   - In this case, we identify `/home/chuckw/public_html/cgi-bin` as the directory that contains CGI scripts for the domain's Web server.

Note

The ability to run scripts on a Web server is often considered a premium service by professional hosting companies. We show you how to let your hosting clients do it here. You may want to leave this line out, however, if you consider it too much of a drain on your server's resources.

   - The `ErrorLog` and `TransferLog` lines let you direct error conditions and file transfer information associated with the domain to their own log files (`example-error_log` and `example-access_log`).

4. Check other settings in the `httpd.conf` file.

   There are a bunch of settings already in place in the `httpd.conf` file. Read through the detailed comments on those settings in the file itself to learn more about what they do. You can (and in fact, you just did) override some of these values for specific virtual Web hosts by adding lines into virtual host containers.

Here are some examples of lines that are set for you (that in many cases you likely don't need to touch).

- **ServerTokens OS** — Prevents visitors from finding out what modules you are running. (You can comment it out to let others see what you are running.)
- **ServerRoot "/etc/httpd"** — Identifies where your configuration files are
- **Listen 80** — Causes the Web server to listen on port 80 for requests to view content
- **User apache** — The Web server (httpd daemon) runs as the user apache.
- **Group apache** — The Web server (httpd daemon) runs as the group apache.
- **UseCanonicalName Off** — Causes Apache to display the host name and port requested by the client when displaying the page. This is what you want with a server doing virtual hosting. The alternative is to always display a fixed server name (set by the ServerName directive) for every page requested.
- **DocumentRoot "/var/www/html"** — Sets the location of your Web site's content. Of course, we override this in our example for each virtual host by using `/home/`*`user`*`/public_html` as the location of each site.
- **UserDir disable** — This prevents the `/home/user/public_html` directory from being shared as `http://`*`domainname`*`/~`*`user`*. In Chapter 11, we change this to `UserDir public_html` since we have only one domain name and it's a good way to let our ISP hosting clients publish personal Web pages.
- **DirectoryIndex index.html index.html.var** — If someone requests a directory, Apache will look for an `index.html` file, then (if that doesn't exist) an `index.htm.var` file to display the contents.
- **AccessFileName .htaccess** — This directive lets you add `.htaccess` files to any directory. Inside that file, you can add directives that determine how the contents of the directory (and all subdirectories) are displayed. While this does give some control to your Web-hosting customers to take some control of how their own content is displayed (without letting them edit the `httpd.conf` file themselves), it can cause slower response than you would get by adding directives directly to the `httpd.conf` file.
- **HostnameLookups Off** — Causes Apache to log information about visitors to the Web site as IP addresses. Change it to On to have Apache look up and display host names instead.
- **ErrorLog logs/error_log** — Causes error conditions to be written to the `/var/log/httpd/error_log` file. We override this for each virtual Web server and create individual error log files.
- **CustomLog logs/access_log combined** — Causes access activity to be written to the `/var/log/httpd/access_log` file. Again, we override this for each virtual Web server and create individual access log files.
- **ServerSignature On** — With ServerSignature set to On (which it is by default), when the server generates a page (such as when there is an error condition), information about the server is written on the page. Information includes the version of the Apache server, the host address, and the port number.

- **Alias /icons/ "/var/www/icons/"** — Sets where Apache will look for icons, in particular when it needs icons for pages and directory listings it generates
- **Alias /manual "/var/www/manual"** — Identifies the location of the Apache manuals, if you install the http-manual package. This provides a great way to search and view the Apache manuals. Just type `http://localhost/manual` in a Web browser from your server.
- **ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"** — Sets the default locations for CGI scripts. We override this in our example to allow each of our hosting clients to add their own CGI scripts. (Using ScriptAlias, instead of just Alias, tells the server that the files in the directory are run as scripts instead of just being displayed.)
- **IndexOptions FancyIndexing VersionSort NameWidth=*** — Sets how indexes of directories are displayed. FancyIndexing displays directory contents, with icons representing each file type and links that let you sort contents by name, last modified, size, and description. VersionSort causes software package to be sorted in a logical order. NameWidth sets the length of filenames that are displayed (*, or no limit, by default).
- **AddType MIME file_extension** — When you add different types of content (document, audio, video, and so on), you can identify what sort of content each is by MIME type and file extension. For example, `AddType application/x-tar .tgz` tells your server to tell clients that files with a .tgz on the end have a MIME type of application/x-tar. With that information, the Web browser client can decide the right thing to do with the content (display it, launch an external viewer, or download it, for example). You can check the `/etc/mime.types` file to see if your content is already defined. If it is not, you can use the AddType directive to add or change MIME and file extension information to properly identify the content you serve.
- **ReadmeName README.html** — Sets the name of the readme file (`README.html`) that Apache will look for when a directory is displayed

> **Note**
>
> Many of the options just shown can be added inside a virtual host container. So you have a lot of flexibility in how you set up each virtual host. Even though they are on the same machine, virtual hosts can behave differently in how they sort data, display files, and log data.

Besides the directives just described, there are a lot of opportunities to tune and enhance your Web server. Here are some examples of features you might want to look into further:

- **Timeouts and servers** — There are a lot of ways you can tune the behavior of your Apache server. You can define how many server processes are running at a time and how servers are started up and stopped to meet load demands. You can also set when various waits for responses can timeout.
- **Modules** — Apache loads various modules to handle special features. For example, there are modules to handle different kinds of authentication, access to databases, and mime types. For more information about modules, see `http://httpd.apache.org/docs-2.1/en/mod`.
- **Languages** — You can indicate the languages supported and their order of preference for your Apache Web server.

5. You can start the Apache server, then set it to start each time you reboot the server. Here's how:

```
# service httpd start
# chkconfig httpd on
```

Step 6: Set up disk quotas

To limit the amount of disk space that each user (or in our case, Web site) can consume, you can use a feature called disk quotas. To use this feature, we are assuming that you created a separate /home partition when you installed Red Hat Linux. If that is true, proceed with the following steps (as root user from a Terminal window):

Note

Before you start, you must have created /home as a separate partition, as described earlier. Otherwise, you can add quotas to the root (/) partition, which is less efficient in our case. If you don't care how much disk space each user consumes, you can skip this whole step.

1. Open the /etc/fstab file in any text editor as root user. Find the partition attached to the /home directory and add usrquota to the comma-separated list of options. Don't change anything else on that line, unless you know what you are doing. For example, if /home were on /dev/hda3, here is how that line might appear after you edit it (the text in bold is what I added):

```
/dev/hda3     /home     /ext3      defaults,usrquota     1 2
```

2. Make sure that no processes have anything in /home open (including any shells). Then unmount and remount /home as follows:

```
# umount /home
# mount /home
```

3. To create an aquota.user file for the /home partition, type the following:

```
# quotacheck -c /home
```

4. To create quotas for a particular user, use the edquota command. (You need to use the vi editor to use this command.) For example, here's how to do it for a user named chuckw:

```
# edquota -u chuckw
Disk quotas for user chuck (uid 502)
   Filesystem   blocks   soft   hard   inodes   soft   hard
   /dev/hda3      5201      0      0       33      0      0
```

This shows that the user chuckw has created 33 files and/or directories on the partition and is currently consuming 5201 blocks of space (about 5.1MB). You can set soft and hard limits on both blocks and inodes. For our purposes, we are going to limit blocks to 300MB soft and 500MB hard. To do that, I changed the last line to appear as follows:

```
/dev/hda3      5201      307200    512000       33       0       0
```

If the user goes over the soft limit, he or she has seven days to go back under that limit or be blocked from writing anything further to the partition. If the user tries to exceed the hard limit, he or she gets an immediate write failure.

5. Create a start-up script for quota check. As root user, create a new file called `/etc/init.d/quota`. Here's an example of what that script can contain:

```
#!/bin/bash
# init file for quota
#
# description: Checks disk quotas
#
# processname: quota
# chkconfig: - 90 90
# source function library
. /etc/rc.d/init.d/functions

case "$1" in
  start)
       echo -n "Checking quotas: "
        daemon /sbin/quotacheck -avug
       echo
       echo -n "Starting quotas: "
        daemon /sbin/quotaon -avug
       echo
       ;;
  stop)
       echo -n "Shutting down quotas: "
       daemon /sbin/quotaoff -a
       echo
       ;;
  restart)
        $0 stop
        $0 start
        ;;
  *)
       echo "Usage: quota {start|stop|restart}"
       exit 1
esac

exit 0
```

6. Make the quota script executable, set it to turn on when you boot, and turn it on immediately as follows:

```
# chmod 755 /etc/init.d/quota
# chkconfig quota on
# service quota start
```

7. To check how much disk space is being consumed by each user in the partition using quotas (in our case, /home), type the following:

```
# repquota -a
*** Report for user quotas on device /dev/hda3
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
User        used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root    -- 364908       0       0           5239     0     0
chuckw --   58716 307200  512000             38     0     0
```

You should now have disk quotas working on your computer for the `/home` directory.

Step 7: Create the firewall

Use the descriptions in Chapter 7 to configure your firewall. Port 80 is the critical port for allowing access to your Web server content. That is the location that Web browsers try by default when simply given a URL for Web content.

Besides port 80, you might consider opening ports 20 and 21 (FTP), 53 (DNS), 22 (remote login with ssh), 25 (e-mail), and port 110 (e-mail retrieval with pop3).

At this point, you should have a working Web server. Next, you can refer to the "Monitoring Your Server" section for information on maintaining your server and keeping it safe. Then, refer to the "Adding Web Server Content" section for information on how your Web-hosting clients can put their content on your Web server.

Monitoring Your Server

Techniques for securing and monitoring your server are mostly the same as those described for the Mini ISP project in Chapter 11. For Web hosting specifically, however, you want to be able to:

- Monitor any patterns of abuse to the Web site.
- Produce statistics to see how much activity each virtual Web site is getting.

Watching log files

Our first suggestion is to watch your basic log files. If you like to do things graphically, Red Hat Linux has a GUI tool for checking your system log files called the System Logs window. To open the System Logs window from the Red Hat menu on the desktop, click System Tools ➔ System Logs (or type `redhat-logviewer`). Figure 12-2 shows an example of this window.

Of interest to you from the System Logs window are the Boot Log (to see if services started up properly), Mail Log (to view e-mail activity), and Security Log (to see who is trying to log in to your system). You should check these files on occasion to make sure that no funny business is going on with the server.

The Apache Access Log and Apache Error Log are most useful for a single domain Web host. You may remember that you created separate access and error log files for each virtual host on your Web server. You can check them the old fashioned way: Open them in a text editor.

If you have the logwatch package installed, there is an easier way to scan through potentially dangerous log messages. Once per day (the default is 4:00 a.m.), logwatch scans through your log files and sends an e-mail to the root user. That e-mail message details any potentially dangerous occurrences (failed login attempts, port scans, and so on).

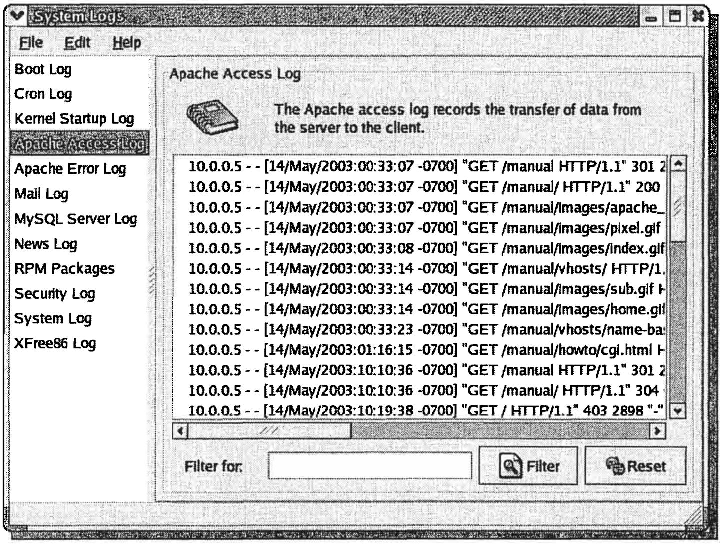


FIGURE 12-2: Watch your system log files from the System Logs window.

You can have that mail sent to any e-mail account that you like by changing the following line in the `/etc/log.d/conf/logwatch.conf` file from root to any e-mail address you want to use:

```
MailTo = root
```

Watching Web site activity

There are lots of tools available for monitoring Web site activity and making colorful graphs to show your clients. One tool for charting such things as the amount of data being transferred and the number of hits you get on a server is called Webalizer. To use Webalizer in Red Hat Linux, do the following:

1. As root user in a Terminal window, type the following:

   ```
   # webalizer
   ```

2. Open Mozilla or another browser window on the server and type the following in the location box:

   ```
   http://localhost/usage/index.html
   ```

   You will see statistics for your Web server.

3. Edit the `/etc/webalizer.conf` file (as root user from a Terminal window). You can configure the file to modify the output in various ways. Refer to comments in the file for information on how to separate information based on different virtual hosts on your Web server.

Adding Web Server Content

We've set up this project so that there is a separate user account set up for loading the content of each virtual Web site. Presumably, that user is the one who is creating the Web content for the site. The following procedure describes how the user can go about uploading content so it is properly installed on the server:

1. **Create content** — Your customers can create any content they want for their Web sites. If they have never created Web content before, you can recommend that they make an `index.html` file, using any HTML composer.
2. **Get an FTP client program** — Sometimes features for publishing Web pages are built into an HTML composer. If those features are not available for the person creating the Web content, he or she needs to get an FTP client program to publish their pages to your Web server.

   If you are composing on a Red Hat Linux system, you can use command-line programs such as `sftp` or `ncftp` to transfer files in FTP. Or use `gftp` to launch a GUI window for transferring files. (To use `sftp` or `ssh`, you need to have the `ssh` service enabled, which it is by default, and port 22 open on your firewall.)

   On a Windows machine, there are shareware or free programs, such as WS_FTP, available to use to transfer Web content to your Web server. (If you don't have an FTP client, you can try `download.com`.)
3. **Copy files to the Web server** — Provide your customers with the user name and password they can use to access the `public_html` directory you set up for their Web sites. Have them copy the content to the `public_html` directory in their home directory after they log in using the FTP client.
4. **Check the content** — If the Web server is running, you can immediately check that the content is available. For example, if the Web site is `www.example.com`, you could type the following into the location box of your Web browser:

   ```
   http://www.example.com
   ```

   You should see the content from the index.html file that was placed in the `public_html` directory for the user associated with that virtual Web server.

Chuck's Tech Tip

While FTP is very useful, it is a protocol that is slowly being made obsolete by the SSH (Secure SHell) protocol. SSH supports direct shell interaction as well as FTP style transfers with a tool called sftp (Secure File Transfer Protocol) and scp (Secure CoPy). The benefit to these tools is that the data you are transferring is encrypted while it is in transit. This prevents third parties from snooping in on the data while it is being transferred. Many people use the excuse that they don't have anything to hide; therefore, they have no reason to use such tools. The problem with that is that their account is a valuable tool to someone with bad intentions. It allows the person to shield his or her identity and cause all of his or her actions to be blamed on *you*! In the absence of encryption, it is trivial to sniff your password and later make use of your account.

Summary

If everything went well, you should now be hosting Web content for several domains. Without adding any software to Red Hat Linux, you could be running your own Web, mail, and FTP services, protected by a firewall. The only thing between you and a real Web-hosting company is that you don't have a beeper alerting you when the network goes down.

Just for Fun

part

V

in this part

Linux on a Floppy and BSD Games

chapter 13

in this chapter

☑ Create a DogHouse Linux bootable floppy

☑ Run some DogHouse Linux shell commands

☑ Play a few BSD games

Even though we said you need at least a Pentium-class computer to do the Linux Toys projects, I imagine someone will be disappointed when they get the book home and nothing works on that 486 computer gathering dust in the closet. If that's you, you'll be happy to know that we have something for you.

Because the latest Red Hat Linux won't install on pre-Pentium-class computers, we have included a little distribution of Linux on the Linux Toys CD that we call DogHouse Linux. You can copy it to a floppy and run it on most computers that have a floppy disk drive. (Yes, it should work on your old 486 machine.)

What you get with our DogHouse Linux distribution is . . . well, not much. But you will get enough to:

- Feel what it was like to use old UNIX systems (when Linus was just a glint in his daddy's eye).
- Try a few classic Linux commands that will work on almost any Linux system.
- Play a few classic pre-Linux character-based games.

If you are truly interested in learning Linux, it won't hurt you to poke around with DogHouse Linux a bit. Even though our DogHouse Linux behaves like an old text-only computer system, the commands it contains are included in even the latest versions of Linux. The key is that DogHouse Linux can be a way to get Linux running on most any PC (see Figure 13-1).

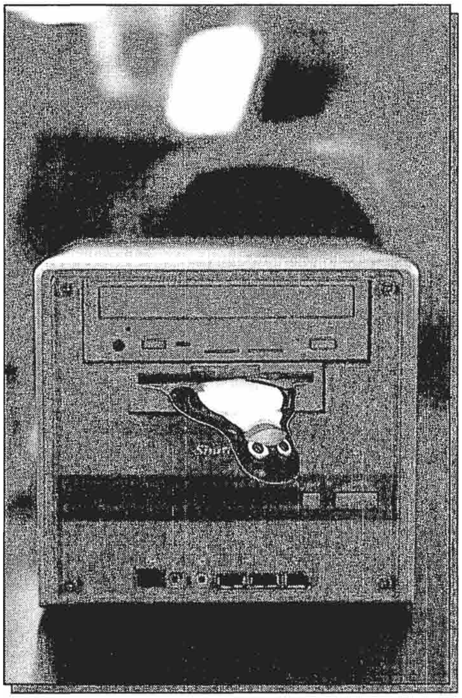


FIGURE 13-1: Run a small Linux distribution from a floppy disk.

Our DogHouse Linux is based on Herbix (`www.cuodan.net/~sina`), a Linux distribution on a floppy that was made to use as a Linux server. Chuck stripped out all the networking features and made it into a Linux distribution with which you could try out a few commands and run some old games.

The games in DogHouse Linux are from a software package called bsd-games. Because the games are all character-based, they don't require a graphical interface — which is good, because DogHouse Linux doesn't have a graphical interface.

This chapter comes in two main parts:

- **Linux Toys DogHouse Linux** — Where you copy DogHouse Linux to a floppy disk (from a file contained on the Linux Toys CD), boot it up, and play around a bit.
- **BSD Games** — Where you play classic pre-Linux games that were originally designed for early UNIX systems. These include games such as fish, wump, arithmetic, and adventure.

Note

If you do have Red Hat Linux installed, you can skip installing DogHouse Linux and still try out the commands we describe here (they're all in Red Hat). Also, we put the bsd-games RPM package on the Linux Toys CD, so you can install and play the complete set of games as well.

Configuring DogHouse Linux

To be able to fit on a floppy disk, DogHouse Linux was created as a 1.4MB file referred to as a disk image. To use DogHouse Linux, you need to copy that 1.4MB image to a floppy disk. Then just stick it into a computer and boot it up. Here's what you do.

Step 1: Gather the hardware

The computer on which you use DogHouse Linux doesn't have to have a CD drive. However, you need to have access to a computer with a CD-ROM drive to be able to create the DogHouse Linux floppy disk. So to get started you need:

- A computer (Windows or Linux) with a CD drive and a floppy drive to create the DogHouse Linux floppy disk
- A blank 1.4MB floppy disk
- A PC (doesn't matter what is on it) with an available floppy disk drive to use the DogHouse Linux floppy

Step 2: Create the DogHouse Linux floppy

Follow this procedure to create your DogHouse Linux floppy:

1. Insert the Linux Toys CD into the Windows or Linux computer.
2. Insert the blank, writable 1.4MB floppy disk into the floppy drive.
3. Copy the DogHouse Linux disk image to floppy disk. How you do that depends on whether you are using a Windows or Linux system:
   - **Windows** — Open an MS-DOS window. Then determine the drive number of your CD drive (possibly D: or E:), and copy the DogHouse Linux disk image to floppy disk. For example, if your CD drive is D:, type this from the DOS window:

   ```
   > cd d:\ch13-DogHouse
   > rawrite -f doghouse.img
   ```

   - **Linux** — Open a Terminal window. If the CD didn't mount automatically, you need to mount it as root user (as shown in the first command that follows). Then you can copy the DogHouse Linux image to floppy disk as follows:

   ```
   # mount /mnt/cdrom
   # cd /mnt/cdrom/ch13-DogHouse
   # dd if=doghouse.img of=/dev/fd0
   ```

4. Pop out the floppy disk and mark it DogHouse Linux.

You should now have a bootable copy of DogHouse Linux. Now you can try it out.

A Lesson in . . . Disk Images

A disk image is a file that has been copied, bit-for-bit, from an entire floppy disk, CD, or even a hard disk partition or other medium. The advantage of this kind of file is that it can be used to easily reproduce the whole contents of the original medium.

These days, floppy disk images are often used when you need to install software on an old computer that doesn't have a CD drive. For example, if you have Red Hat Linux on CD, the `images` directory on the first CD contains floppy disk images you can use to install Red Hat Linux from over a network.

If you have a high-speed (such as DSL) connection to the Internet, you can download entire CD images in a few hours. CDs are often stored as ISO images, which refers to the ISO-9660 standard that defines their structure. CDs in this format can be used on Linux, Windows, and most modern operating systems.

With a copy of a disk image on your computer, you can mount that disk image (provided it's in a file system structure known to Linux) to view its contents. For example, here's how you could copy and mount a boot disk image from the first Red Hat Linux installation CD so that you could view its contents from Linux. Insert that CD and do the following as root user from a Terminal window in Red Hat Linux:

```
# mount /mnt/cdrom
# cp /mnt/cdrom/images/bootdisk.img /tmp
# mkdir /mnt/bd
# mount -o loop /tmp/bootdisk.img /mnt/bd
# ls /mnt/bd
```

The `ls` command will show you the contents of the `bootdisk.img` file, as they would appear on the floppy disk. You can do this kind of `loop` mount using any kind of disk image that Linux can identify. When you are done, leave the directory and type `umount /mnt/bd`.

Running DogHouse Linux

One of the cool things about DogHouse Linux is that you can run it on most any computer that lets you boot from a floppy disk. The DogHouse Linux floppy is actually a bootable computer operating system. Any operating system you have on your hard disk is ignored, and the contents of the floppy disk take over and run your computer.

Using DogHouse Linux

While it doesn't have much in it, DogHouse Linux does let you try out some Linux commands on an older, less powerful computer that's not up to the task of running Red Hat Linux. To get started with DogHouse Linux, do the following:

1. Insert the DogHouse Linux floppy disk you created in the previous section into the disk drive and reboot the computer.

Note

If the floppy disk doesn't boot, you may need to check your computer BIOS. Reboot and when the POST screen shows a message telling you how to go to Setup Mode, quickly press the key it mentions (possibly F2 or Delete). Look for an entry that shows boot order and make sure that the boot order includes booting from floppy disk before hard disk.

2. After DogHouse Linux comes up, type `root` and press Enter when you see the login prompt.

DogHouse Linux will run in your system's memory and not disturb the contents of your computer's hard disk in any way. You can safely play the games and run commands. Then just pop out the floppy and reboot to restart your computer as it was.

Chuck built DogHouse Linux from the Herbix project (`www.cuodan.net/~sina/`). Inside of Herbix is a program called busybox (`www.busybox.net`). This neat little utility implements most of the basic Linux shell commands used in DogHouse Linux. The short Linux shell tutorial in this section centers on the commands included in busybox.

Cross-Reference

For more information about using shell commands, see Appendix B.

Here are a few commands you can try in DogHouse Linux:

1. Create a directory called `/tmp/test`:

```
$ mkdir /tmp/test
```

2. Create a file called `hellothere` by echoing the set of words shown here to that file:

```
$ echo "echo my name is \$USER" > /tmp/test/hellothere
```

3. Show the contents of the `hellothere` file:

```
$ cat /tmp/test/hellothere
echo my name is $USER
```

4. Show the permissions associated with the `hellothere` file. The following example shows that the `hellothere` file has read/write permission for the owner of the file (rw-) and group (rw-) assigned to the file. All others have read permission (r--). The owner and group are both root, represented by UID and GID 0. The file has 22 characters in it.

```
$ ls -l /tmp/test/hellothere
-rw-rw-r--    1  0  0    22   /tmp/test/hellothere
```

5. Change the permission of the file to make it executable and able to run as a command. For example:

```
$ chmod 755 /tmp/test/hellothere
$ ls -l /tmp/test/hellothere
-rwxrwxr-x    1  0  0    22   /tmp/test/hellothere
```

6. Next, run the file as a command. You see that the words we put in the file consisted of the `echo` command, followed by some text that will be echoed to the screen. When the command runs, $USER is translated into your user name:

```
$ /tmp/test/hellothere
my name is root
```

7. Try a couple of the games now, if you like.

Playing games on DogHouse Linux

There are five games on the DogHouse Linux floppy. When you first start DogHouse Linux and log in, you are dropped into the `/usr/local/games` directory. From there you can play the following games:

- **adventure** — Explore the Colossal Cave to bring back treasure, taking care to avoid dwarves with axes and pirates who will steal and hide your treasure deep in the caves.
- **wump** — Travel through tunnels and shoot the Wumpus with arrows before he eats you.
- **morse** — Translate words and letters into Morse code (dit, daw, dit, daw).
- **fish** — Play Go Fish against the computer.
- **arithmetic** — Add and subtract numbers as quickly as you can.

To play any of those games in DogHouse Linux, just type the command shown. To quit a game, hold the Ctrl key and press D (Ctrl+D). Of the few games included in DogHouse Linux, `adventure` is the most interesting. The other games we describe here are `wump` and `fish`.

Playing adventure

The first computer game I can remember playing was adventure. I was attending Syracuse University and I got a login account to write some BASIC programs. At night, system administrators would open permission on the games to let us play.

Because our computer terminals consisted of typewriter-like, scrolling paper devices, it was easy to take our game output back to the dorm and work on our maps of the Colossal Cave.

In adventure (`adventure` command), you type one- or two-word commands to move around the caves, pick things up, put things down, and look around. You tell your guide what to do. Your guide tells you where you are and what items are there.

You start above ground, outside a small brick building. Your goal is to explore underground caves and bring back as much treasure as you can, without being killed.

Here are a few tips to get you started:

- If you don't know what to do first, type the word `building` to go inside the small brick building. Type `take lamp` (you aren't going anywhere in the caves without a lamp). Type `light lamp` once you are in the caves.

- To get into the caves, you can walk around outside until you find a grate and open it with the keys you take from the building. From inside the building, use secret words `xyzzy` or `plugh` to magically transport yourself to and from the building and particular rooms in the cave.
- Move around the caves in adventure by asking for an adjacent room, by naming a direction (N, S, E, SW, NE, and so on), or by asking to go up and down.
- As you explore the caves, draw a map as you go — it helps you return to places later.
- Take valuable things and bring them back to the building.
- Watch out for dwarves. If you find an axe, you should suspect a dwarf is close by. (Take the axe — you may need it!)
- If you get stuck in twisty little passages, immediately drop an item. That way you will know when you are back in the room where you fell into the passages.
- You need a cage to catch the bird, but you won't catch him if you are carrying a rod.

The further you explore into the caves and the more treasures you bring back and drop in the building, the higher your score will be. Type `help` for further hints on what to do. Here's what a bit of adventure looks like:

```
$ adventure
Welcome to Adventure!!  Would you like instructions?
n
You are standing at the end of a road before a small brick
building.
Around you is a forest.  A small stream flows out of the
building and down a gully.
building

You are inside a building, a well house for a large spring.
There is a shiny brass lamp nearby.
take lamp
OK
xyzzy
It is now pitch dark. If you proceed you will likely fall into
a pit.
light lamp
Your lamp is now on.
You're in Debris Room.
A three foot black rod with a rusty star on an end lies nearby.
w
You are in an awkward sloping east/west canyon.
w
You are in a splendid chamber thirty feet high.  The walls are
frozen rivers of orange stone. An awkward canyon and a good
passage exit from east and west sides of the chamber.
A cheerful little bird is sitting here singing.
```

Playing wump

Hunt the Wumpus (wump command) is another game you play in a cave. There are twenty rooms in the cave, with each room having three tunnels connecting it to other rooms. Your job is to find the Wumpus and shoot him with your arrows.

As you move between the rooms, watch out for pits (if you fall in, you die). Also, avoid close contact with a Wumpus itself, since it will eat you. Bats are less dangerous. If you run into some bats, they simply take you to another part of the cave.

Your two choices of action are move (m) or shoot (s). Type a room number with the `m` command to move to that room (such as `m 7`). Indicate one or more rooms when you shoot to set the path of your crooked arrow. Here's how it feels to fall in the pit:

```
# wump
Instructions? (y-n) y

You're in a cave with 20 rooms and 3 tunnels leading from each room.
There are 3 bats and 3 pits scattered throughout the cave, and your
quiver holds 5 custom super anti-evil Wumpus arrows.  Good luck.

You are in room 3 of the cave, and have 5 arrows left.
*rustle* *rustle* (must be bats nearby)
*whoosh* (I feel a draft from some pits).
There are tunnels to rooms 1, 4, and 5.
Move or shoot? (m-s) m 4
*AAAUUUUGGGGGHHHHHhhhhhhhhhh...*
The whistling sound and updraft as you walked into this room of the
cave apparently wasn't enough to clue you in to the presence of the
bottomless pit.  You have a lot of time to reflect on this error as
you fall many miles to the core of the earth.  Look on the bright side; you can
at least find out if Jules Verne was right...

Care to play another game? (y-n) n
```

Playing Go Fish

With the `fish` command, you can play the classic Go Fish card game. Just type `fish` and ask the computer for a card. Get more sets of four cards than the computer and you win. Here's an example:

```
$ fish
Would you like instructions (y or n)? n
I get to start.
I ask you for: 2.
You say "GO FISH!"
Your hand is: 5 7 9 9 10 J J
You ask me for: 9
```

Running BSD-Games

If you used a UNIX computer in the early 1970s, there were no windows, menus, or icons to smooth your trip. You typed to a computer on a scrolling roll of paper or, if you were lucky enough to get access to one, a text-only green screen. That was the environment in which the first computer games were created.

Games such as adventure, quiz, and fish challenged your mind without assaulting your senses. These same games are out of the mainstream but are still around today — available most commonly in a software package called bsd-games.

As I described previously, we took a few games from the bsd-games package and placed them on the DogHouse Linux floppy image. If those few bsd-games weren't enough for you, we have the full package of bsd-games on the Linux Toys CD that you can try out. The bsd-games package includes the five games we have on DogHouse Linux but has about 30 more games. To play them, you can bypass DogHouse Linux and simply install the bsd-games package on the same Red Hat Linux system you use for all the other projects in this book.

Here's how to install the full set of bsd-games on a Red Hat Linux system:

1. Insert the Linux Toys CD into the CD drive of a running Red Hat Linux system.
2. If it doesn't mount automatically, type the following as root user from a Terminal window:

   ```
   # mount /mnt/cdrom
   ```

3. Type the following to install the bsd-games packages:

   ```
   # rpm -Uhv /mnt/cdrom/ch13-DogHouse/bsd-games*
   ```

If you installed the bsd-games package in Red Hat Linux, you should add `/usr/games` to your PATH by adding the following line to your `$HOME/.bashrc` file (using any text editor):

```
export PATH=$PATH:/usr/games
```

Then you can type the name of the game you want to play. Table 13-1 shows the games that come with the complete bsd-games package included on the Linux Toys CD.

Table 13-1 Character-based BSD Games

Game	*Description*
adventure	Explore the Colossal caves and find treasures
arithmetic	Answer simple, timed arithmetic questions
atc	Air traffic controller game (direct jets and propeller planes through flight area and airports)
backgammon	Backgammon board game

Continued

Table 13-1 *(continued)*

Game	*Description*
battlestar	A tropical adventure game
bcd	Take any string of text and output it to punched card format
caesar	Decrypt caesar ciphers
canfield	The canfield solitaire game
cribbage	The cribbage board game
dm	Dungeon master to regulate game playing
factor	Factor numbers
fish	The Go Fish card game
gomoku	Two-player game to get five in a row
hunt	Multiplayer maze game with the object of killing the other players
mille	The Mille Bornes game
monop	The Monopoly game for multiple players
morse	Translate text into morse code
number	Converts Arabic numbers to words
phantasia	Role playing fantasy game for fighting monsters
pig	Reformat input as pig latin
pom	Displays the current phase of the moon
ppt	Translate text into paper tape output
primes	Generate prime numbers
quiz	Answer questions on selected topics
rain	Displays animated raindrops
random	Generate random numbers
robots	Fight evil robots
rot13	Create caesar cyphers rotated by 13 characters
sail	Multiuser fighting ships game
snake	Avoid getting eaten by a snake
snscore	Check the scores from snake games
teachgammon	Learn backgammon rules and controls
tetris-bsd	Tetris game

Game	Description
trek	Trekkie space game
wargames	No thermonuclear warfare (not dangerous)
worm	A growing worm game
worms	Animated worms
wump	Shoot at the Wumpus in an underground cave

Trying Other Bootable Linuxes

DogHouse Linux is just one of many Linux-on-bootable-media distributions available today — and not one of the most interesting to advanced users. If you find the idea fascinating of bootable operating systems on a floppy or CD-ROM, there are ways you can look into other special-use bootable Linux distributions.

One great place to look for bootable Linux distros is `Distrowatch.com`. The `Distrowatch.com` site contains tons of Linux distributions that you can download and try out. Some of these are made to simply boot from a floppy disk or CD-ROM and run. Others are full-blown, installable Linux distributions.

As for individual bootable Linux distributions, here are some that might interest you:

- **GeeXboX** (`http://geexbox.free.fr`) — You can play digital media from your hard disk by booting a GeeXboX CD-ROM. You can watch movies from DVD (if you have a player) or VCD or play audio CDs in a variety of formats. The advantage of GeeXboX is that you don't have the overhead of a full-blown Linux distribution just to play video or music.
- **Bootable Business Card** (`www.lnx-bbc.org`) — For a bootable Linux system that you can fit in your wallet, there is the Bootable Business Card. This Linux distribution fits on a mini-CD that is in the shape of a business card. It can be used as a rescue CD, to boot a computer and get to your hard disk if software repairs are needed. It even includes X, so you have a GUI to work with.
- **Coyote Linux** (`www.coyotelinux.com`) — If you want to share an Internet connection through a single computer, Coyote Linux is a simple way to make a computer into a router/firewall. Coyote Linux supports dial-up or Ethernet Internet connections. It can run on hardware as old as a 486 with 12MB of RAM.

As I mentioned earlier, Herbix itself is also a good bootable Linux. A big advantage of Herbix is that you can use it to turn a computer that is loaded with your Web server content into a Linux server.

Summary

Using Linux from nothing more than a shell with just a few commands can give you the feel of what it was like to use early UNIX systems. Even with our Linux Toys DogHouse Linux on a floppy and a very old computer, you can run some basic commands to work with files and get around the file system. Playing some classic, character-based games can further give you a feel for what old-time computing was like.

Controlling Toy Cars

chapter 14

in this chapter

- ☑ Controlling toy cars from Linux
- ☑ Configuring your toy cars
- ☑ Operating your toy car manually
- ☑ Running your toy car in patterns
- ☑ Racing your toy cars over the Internet

This project is more "Toy" than "Linux." The idea is to be able to take inexpensive remote control (RC) cars and operate them from Linux. Although it's sort of a dumb thing to do (if you're standing there, it's a lot easier to just press the buttons on the remote control), wiring an RC transmitter to our Linux box lets us illustrate a few cool things:

- **Run courses** — Using signals that turn relays on and off, Chuck created some simple scripts (written in the Perl scripting language) that cause the car to run in set patterns around a room.
- ***Really* remote control** — Once you can control the car from Linux, with an Internet connection you can control the car from anywhere in the world. Add a video camera (using the same procedure for setting up streaming video described in Chapter 8), and you can have multicar remote control races as you operate one of the cars.
- **Start unattended races** — Using the Linux cron facility, you can set your cars to start running automatically at any time you set. This is a great feature if you want to, for example, freak-out your dog in the middle of the night. (See Chapter 9 for a description of how to use cron.)

There are also some hardware and software features you can explore along the way:

- **Use relays** — With the LynX-PORT controller, you can refine how to turn relays on and off from Linux. In this case, the relays can control how the car turns, goes forward, and backs up.
- **Play with serial ports** — Because the LynX-PORT controller connects to a serial port (also referred to as a COM port on a PC), you can learn something about how serial ports work by sending signals through them.

The electronics part is pretty simple (challenging only if you're not used to soldering): Solder wires to the remote control and connect them to the LynX-PORT controller. Once you know what each relay does, you can play with the scripts we provide or send signals to the cars (as we describe later) to get the cars moving around the way you like.

Chuck's Tech Tip

The larger purpose of this project is to provide a proof of concept and see where it takes us. Problems such as positional feedback and obstacle avoidance make the project an interesting one to tackle. Consider enhancements such as video and audio feedback in order to create a roving security sentinel for your house. Perhaps you're away from your house and you get a call from your alarm company informing you of a security concern at your house. You could use an Internet connection to direct your sentinel to view and record the source of the alarm.

About Controlling Toy Cars

There are hundreds of different kinds of radio control (RC) toys available today— besides cars, there are toy boats, airplanes, blimps, motorcycles, and even turbine-powered jets. We thought it would be fun to connect some RC cars to a computer and illustrate how you can then use Linux to control the cars.

The concept is to open up the RC toy's controller and then connect it into a computer running Linux to operate the controller. We used a LynX-PORT board to connect our toy car controller in such a way that it could be plugged into a serial port on our Linux machine.

With a PC, the port board, a few parts, and a bit of soldering, we were able to get the toy car running from Linux in not much time.

Configuring the Toy Car Controller

The general steps for configuring the toy car controller are as follows:

1. Gather hardware.
2. Install Linux and Toy Car software.
3. Wire up the controllers
4. Test your car.

Step 1: Gather hardware

The main hardware items you need for the project are a PC, a controller board, and a toy car. You need a few tools and other items (listed following) to connect it all together as well.

A few tools

This may be a good opportunity to add a few items to your tool collection. To operate the toy car controller, you are going to solder some wires into the controller and attach those wires into the LynX-PORT board. The tools you need to get that done include:

- **Soldering iron** — To solder wires to the controller. (Most any soldering iron 30W or less and intended for electronics will do.)
- **Soldering kit** — To help you solder. The kit might include solder, heat- sync clamp, fine-point probe, slotted point probe, brush, scraper, soldering gun holder, and desoldering tools. Use thin, lead-free, rosin core solder.
- **Goggles** — To protect your eyes as you solder
- **Small screwdrivers** — To take apart the toy car transmitter
- **Wire strippers** — To strip the wires needed to connect the controller to the ports board
- **Clamp tool or clips** — To hold the controller while you work on it
- **Continuity tester** — To test the connection to the controller
- **Drill (or Dremel tools)** — To drill holes in the controller for wires to come through

Figure 14-1 shows the tools we used.

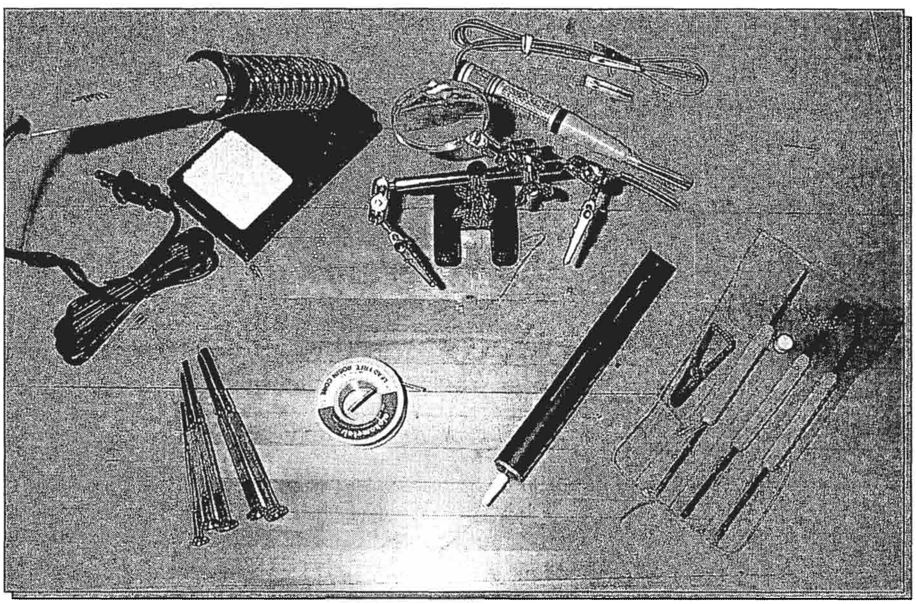


FIGURE 14-1: Pick up the tools you need from your local hardware store.

A few pieces

You need a few extra pieces to finish off the project:

- **Some wire** — To connect the toy car controller to the LynX-PORT board. We just used a few feet of Category 5e wire. The four twisted-pair wires could be separated to make the eight connections we needed.
- **RS232-C serial cable** — To connect the LynX-PORT board to a COM port on the computer
- **Power supply** — To power the LynX-PORT board, get a 9V-12V AC or DC wall-mounted transformer or an external power supply.
- **Plexiglass, mounting screws, and plastic feet** — To connect to the LynX-PORT board so it doesn't slide around

The personal computer

For this project, you can use the minimum personal computer described in Appendix C. The only components the PC needs are:

- **Hard disk** — Large enough to do a minimum Red Hat Linux installation (about 475MB of free hard disk space)
- **CD-ROM drive** — To install Red Hat Linux and the Linux Toys software
- **COM port** — To connect to the LynX-PORTS board (most PCs have at least one)
- **Network card or modem (optional)** — If you want to run races over the Internet (or other network)
- **Video capture card (optional)** — If you configure the project to be able to watch your races over the Internet (or other network). (See Chapter 4 for recommended video capture cards.)
- **Video camera (optional)** — Again, to have Internet car races

A controller board

It's a bit of overkill for running a toy car, but as our tech editor, Kevin Pedigo, said in the description of the RPM for this project, "We blew a couple hundred bucks on a really nice X.10 control board and this is the coolest thing we could think of. Don't blame us. It was a long day."

Anyway, the LynX-PORT multipurpose I/O board from Marrick, Inc. (`www.marrickltd.com`) is made to control a variety of low-voltage devices from your computer. The board works with X.10 security systems and X.10 motion sensors. It can be used to manage a variety of devices, including audio components, garage doors, sprinkler systems, thermostats, and other equipment.

Using the board's four analog inputs, you can monitor current light, moisture, and temperature conditions. Eight digital inputs can monitor motion sensors, magnetic contacts, and various user inputs. There are lots of possibilities with this board.

For this project, we use the board's eight programmable DPDT relays to control the toy cars. Essentially, we are turning the relays off and on to make the cars go left, right, forward, and back.

We found the board for $239 at Smarthome.com. As we said, way overkill for a toy car project. But once you begin playing with the controls for turning off and on relays from Linux, you can use the board for turning off and on all kinds of devices (a model train is the next project we have in mind).

Note

If you use the board from Linux for any home-control applications, contact us at `LinuxToys.net`. We want to help you and others get the most out of the $239 investment.

The toy cars

We used several different types of cars with this project. Many toy stores and electronics stores carry RC cars. Your local Radio Shack carries popular micro RC cars (under the ZipZaps brand), as well as a variety of larger cars and trucks.

We first tried the project with ZipZaps. From there we expanded to some other RC vehicles, which all worked pretty much the same. Here are some issues related to choosing RC cars (or other toys) for this project:

- **Frequencies** — Each remote control toy operates at a particular frequency. This is important, because if you want to run two or more cars at once, they need to use different frequencies. Common frequencies are 27 MHz and 49 MHz. However, you can also find cars that operate at 35 MHz, 45 MHz, 57 MHz, or other frequencies.
- **Number of controls** — The number of controls determines the number of relays you need to use from the ports board. For example, the ZipZaps cars have four controls (forward, back, right, and left). Mini Transbots have two (forward and stand/spin around). Because the LynX-PORT board has eight analog ports, you can connect eight controls. That would let you connect two ZipZaps or four Mini Transbots.
- **Batteries** — Both the cars and the transmitters require batteries. ZipZaps are recharged from the transmitter itself but run for only a few minutes. Longer battery life is crucial to having effective races with the cars.

We went to our local Radio Shack and picked up a few different RC cars. The ZipZaps (the smaller car) we thought was really cool, so we used it for our examples of running scripted patterns. Because the Mini Transbot has only two controls (forward and stand/turn around) and longer battery life, it is a better choice for multicar races or obstacle courses. Any RC car, however, will work fine for this project. Figure 14-2 shows two cars we bought at a local grocery store.

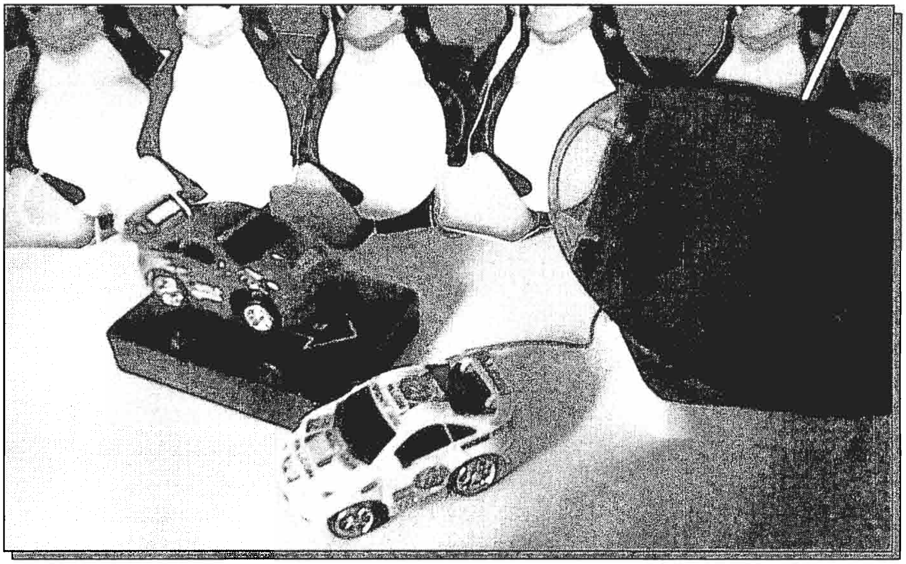


FIGURE 14-2: For this project, we picked up a few toy cars at the grocery store.

Chuck's Tech Tip

Any RC that has 10 or fewer toggle or rocker switch type controls will work for this project. More advanced RC cars have analog slider type switches that allow you to adjust the intensity of a particular setting such as speed or degree of turn. These types of controls will not work with the project in its current incarnation. The LynX-PORT board does have analog style outputs that could work just fine for this purpose with a little hacking.

Step 2: Install Red Hat Linux and toy car software

For the basic Car Controller project, you can use any installed Red Hat Linux system. Once Red Hat Linux is installed, you must install the ltrc RPM package from the Linux Toys CD. If you do a minimum install, you need to add the following packages from Red Hat Linux:

- perl-TermReadKey
- perl-Time-HiRes

We put those packages on the Linux Toys CD for convenience. To install those and the ltrc package, insert the Linux Toys CD and type the following commands as root user from a Terminal window:

```
# mount /mnt/cdrom
# cd /mnt/cdrom/ch14-RemoteControl
# ./install.me
```

To do the optional Internet car race, you need to follow installation instructions from Chapter 8 so you have the Linux installation you need to set up streaming video.

Step 3: Wire up the controllers

You need to take apart your toy car controllers and do a bit of soldering here. The idea is to have two wires connected to each set of relays (eight wires for a ZipZaps and four wires for a Mini Transbot, total). This procedure steps you through the process of wiring a ZipZaps controller and connecting it to the LynX-PORT board.

This procedure could result in irreparable damage to your controller and will certainly void your warranty. Please don't use a car that is a child's favorite, a family heirloom, or worth thousands of dollars. In other words, get a cheap car and expect that you might break it!

1. **Prepare wire** — Get eight strands of wire (for a ZipZaps car) about eight inches in length. We used a length of Category 5e wires and untwisted the four pairs of wires. Strip about ¼" off each end of each wire.
2. **Open controller** — Unscrew the back of the toy car controller. (Watch for springs that might pop out.)
3. **Identify leads** — On the car's remote control board, identify the location of the buttons used to control the car. Look for the two leads going to each button. Figure 14-3 shows a toy car controller with one button removed to show its leads.

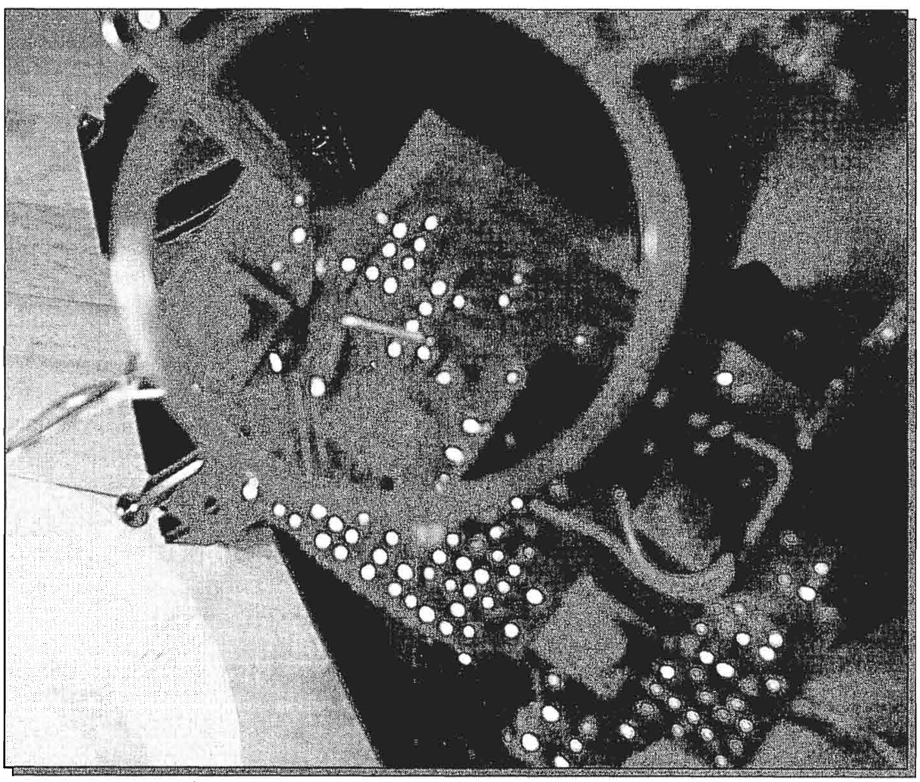


FIGURE 14-3: Identify the leads going to each button.

4. **Drill holes** — Using a Dremel tool (or other drill with a small drill bit), do the following:
   - Drill one hole for each button, so two wires can be pulled through each. Figure 14-5 shows where we chose to drill each hole.
   - Drill eight neat holes across the back of the controller case, so that you will be able to pull the wires through.
5. **Solder wires** — Pull a set of two wires through a hole you drilled, and solder them on to the two leads that complete the connection for the selected button. Repeat that for each button. If you don't know how to solder, we're not going to teach you. However, check out Chuck's Tech Tip on soldering, just up ahead, for a few tips.

   On the ZipZaps, you have to do some pretty fine soldering. If you use four sets of twisted pair from a Category 5e cable, you can solder the wires on as shown in the following table.

	Forward	*Back*	*Left*	*Right*
WIRE	or/wh wh/or	bl/wh wh/bl	br/wh wh/br	gr/wh wh/gr
RELAYS	5A-C 5A-NO	6A-C 6A-NO	7A-C 7A-NO	8A-C 8A-NO

Note If you are using a different RC car, try to determine which buttons on the car controller make the car go forward, back, left, and right. Then connect the wires to the same relays described for the ZipZaps.

6. **Test continuity** — Use a continuity tester to check that each pair of wires can create the connection needed for each button.
7. **Pull wires through** — Gently pull the wires through the back of the controller case. Figure 14-4 shows a controller with wires coming through the back.
8. **Connect wires** — Connect each wire to the LynX-PORT board. If you soldered the wires on as described two steps earlier, connect the wires to the relays on the board as shown in Figure 14-5.
9. **Connect RS-232C cable and power supply** — Connect the RS-232C cable (standard serial cable) between the LynX-PORT board and a COM port on the back of the computer. Also, plug in the power supply to the board.

   The scripts used in this project expect you to connect the cable to the COM1 port (represented by the `/dev/ttyS0` device in Linux). If you use a different port, you must edit the scripts described later and change `/dev/ttyS0` to `/dev/ttyS1` (for COM2) or `/dev/ttyS2` (for COM 3).

At this point, you should have everything ready to start testing your Linux Toys toy car controller.

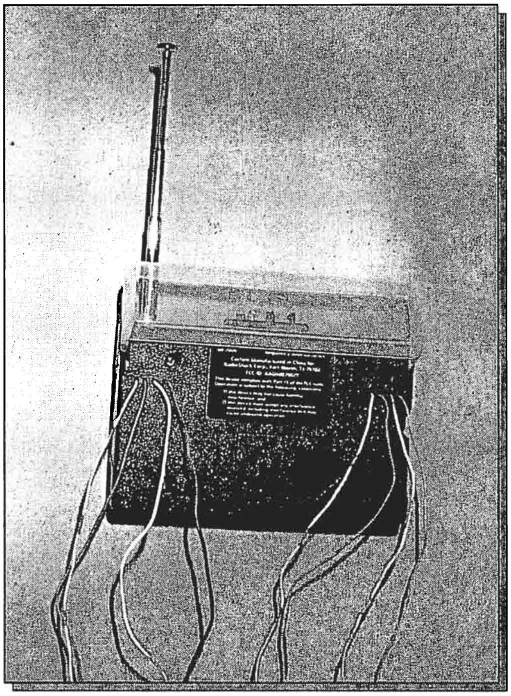


FIGURE 14-4: Pull wires through the back of the controller.

Chuck's Tech Tip

If you're an old hand at soldering, you can probably ignore this tip. If you haven't soldered before, here are some things to take into account.

Soldering pens are *hot*. They have a bad habit of letting smoke out of important components. Use some sort of clamp-on heat-sink tool to bleed away excess heat. You can find just about any soldering tool you need at your local Radio Shack. As for actual soldering technique, avoid touching the solder with the tip of the soldering pen. You get the best solder seal by heating the metal contacts you are trying to solder and letting the solder flow into the area that has been heated.

It is also helpful to use some light sandpaper periodically to rid your soldering tip of soot and grime. Once you've cleaned it, coat the tip with some solder to keep it clean. Above all, if you've never soldered before, practice first. Get some loose wire and solder it to some old circuit board until you're comfortable with the process.

Step 4: Test your car

Turn on your toy car controller and charge up your car. When the light turns green, you are ready to go. Try the buttons on the controller manually (they should still be working) to make sure that the controller can communicate with the car. You are now ready to test the car out from Linux.

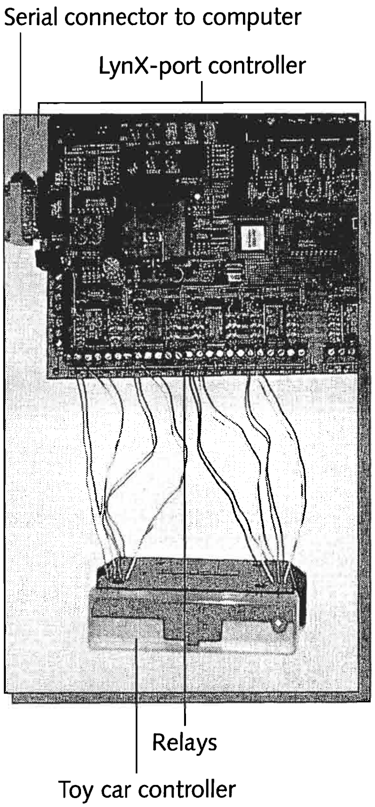


FIGURE 14-5: Connect wires from the controller to the LynX-PORT board.

Operating the Car Manually

To check if your car is working from Linux, log in to Linux and open a Terminal window. From the Terminal window, you can run the `ltrc-manual.pl` script as follows:

```
$ /usr/local/bin/ltrc-manual.pl
```

With the script running, use the keyboard to go forward (K), back (J), left (H), right (L), and stop (N). The LynX-PORT board should make click-clack noises with each key press. Lights for each relay should go on and off. If your car moves around as it should, type Ctrl+C to exit from the script. Next, try running some preset patterns.

Note

If the script fails to open the COM1 port, open a Terminal window and type `su -` and the root user password (when prompted), and type `chmod 666 /dev/ttyS0` to open permission to your COM1 port.

Operating Your Car in Patterns

You can use the `ltrc-auto.pl` script to operate the car in preset patterns. Three simple patterns come with the ltrc package:

- Circle (`circle.lxp`)
- Figure Eight (`figure8.lxp`)
- Square (`square.lxp`)

You can run any of those patterns using the `ltrc-auto.pl` script. For example, to run the square pattern, you would type the following from the Terminal window:

```
$ /usr/local/bin/ltrc-auto.pl square.lxp
```

Create your own pattern files if you like and put them in the `/usr/local/bin` directory. Then replace `circle.lxp` with the name of your pattern file. Pretty dumb, but kind of cool too.

The pattern scripts consist of a list of commands. Each command tells the car which way to go and how long to go that way. The commands are F (forward), B (back), L (left), R (right), FL (forward-left), FR (forward-right), BL (back-left), and BR (back-right). The number that follows represents the duration that the relay stays on (in milliseconds). The `figure8.lxp` pattern goes forward-left 2.2 seconds, the forward-right 2.2 seconds:

```
FL 2200000
FR 2200000
```

The `square.lxp` script goes forward and forward-left at increments between .3 second and .6 second:

```
F 600000
FL 600000
F 400000
FL 600000
F 400000
FL 600000
F 300000
FL 350000
```

The `circle.lpx` script simply goes front-left for 2.2 seconds.

```
FL 2200000
```

You can create scripts that do any pattern you like. Just create a .lpx file, include the directions you want, and copy it to the `/usr/local/bin` directory. Try making one that drives around where your cat is sleeping and comes back to you. Or try to make it drive around the edge of your table without falling off.

Controlling Races Over the Internet

By now you should have the basic concept of the project. Our next job is to set up the cars so they can be used to race or run obstacle courses over the Internet (or another network). In this case, we want to:

- Use two cars instead of one (so they can compete or bash into each other or whatever). Make sure that they use different frequencies!
- Use cars with longer battery life (ZipZaps aren't made to run very long).
- Set up user accounts so someone can log in over a network to run the cars.
- Set up a camera on the racecourse so the drivers can see their cars.

As I mentioned earlier, I used a car called a Mini Transbot that I got from Radio Shack. The car has only two buttons, so I needed only to use the first two sets of relays. For the second car, I used the second set of relays (5 through 8). The scripts we provide allow you to use two- or four-button controllers. Here's what you do:

1. Follow the Configuring Toy Car Controller procedure to add your first car to the LynX-PORT board (probably replacing the ZipZaps with a car that has longer battery life). If the car has only two buttons, connect only two sets of wires (to relays 1A and 1B).
2. Add a second car to the second set of relays (relays 5 through 8), using the relays shown in the accompanying table.

	Forward	*Back*	*Left*	*Right*
WIRE	or/wh wh/or	bl/wh wh/bl	br/wh wh/br	gr/wh wh/gr
RELAYS	5A-C 5A-NO	6A-C 6A-NO	7A-C 7A-NO	8A-C 8A-NO

3. Configure a video camera (as described in Chapter 8) to provide streaming video from your toy car controller computer.
4. Create a user account that uses the `ltrc-manual-car-1.pl` script as its login shell. Here is an example, as root user from a Terminal window, of creating a user account called `car1`:

   ```
   # useradd -m car1 -s /usr/local/bin/ltrc-manual-car-1.pl
   ```

5. Repeat the previous step, but this time create a user named car2 and use the `ltrc-manual-car-2.pl` script as follows:

   ```
   # useradd -m car2 -s /usr/local/bin/ltrc-manual-car-2.pl
   ```

6. Create passwords for car1, and then repeat the step for car2 (adding passwords as prompted):

   ```
   # passwd car1
   Changing password for user car1.
   New password: *****
   Retype new password: *****
   ```

7. Now, this step is a big leap. The computer must be accessible from the Internet (or other network) and allow:
   - **Login service from the network.** Refer to Chapter 11 for information on how to do that.
   - **Streaming video from the network.** Refer to Chapter 8 for information on how to do that.
8. Have two different people connect to your computer by:
   - **Opening up a video player to your streaming video**. Again, refer to Chapter 8 for further information.
   - **Logging in to your computer** (car1 and car2).

   Instead of getting a shell prompt, users can immediately begin using the following keyboard keys to control their cars:

   ```
   H   Left
   J   Backward
   K   Forward
   L   Right
   N   Stop
   ```

   The streaming video should allow users to watch the race as it progresses (a bit of a time lag should only make it more interesting). Figure 14-6 shows an example of a race in progress.

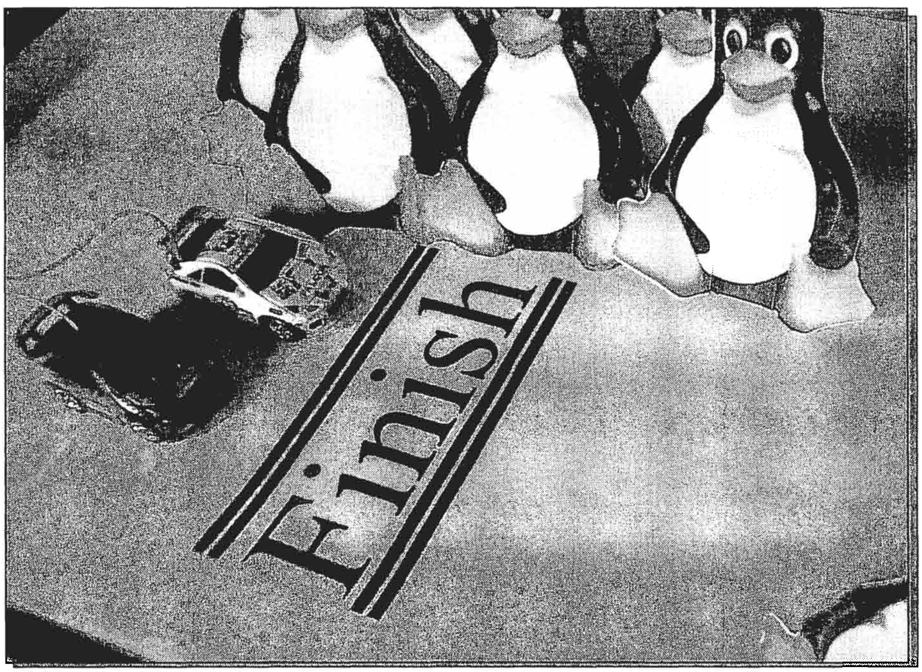


FIGURE 14-6: Every Linux toy car race should end at the "Finnish" line.

Controlling Blimps, Planes, and other RC Toys

There is a wide range of radio-controlled devices you could use with this project to control from Linux. To get more ideas, check out these Web sites:

- **Remote Control Cars and Boats** (`www.remotecontrolcarsandboats.com`)
- **Radio Shack** (`www.radioshack.com`; click the Electronic Kits, Games and Toys link)
- **Radio Control Toy Store** (`www.radiocontroltoystore.com`)

You might want to steer clear of the more expensive toys, unless you have a lot of faith in your soldering ability.

Summary

There's something cool about hitting some keys from your computer and making toy cars run around. By adding a video camera and special Linux Toys toy-car logins, you can have people operate and watch races over the Internet or another network. Also, once you get the idea of how to use and control devices from a relay board, like your toy cars, you begin to see lots of other possibilities for operating devices from that board.

chapter 15

Creating a Digital Picture Frame

in this chapter

☑ Choosing hardware

☑ Installing Red Hat Linux and Toys software

☑ Adding digital images

☑ Dismantling the laptop

☑ Installing components into the picture frame

Once you buy a digital camera and some memory, taking tons of pictures is easy. What you need now is a way to display tons of pictures and still leave room on your coffee table for, well, coffee.

The cost of old laptop computers has come down to the point where you can pick up a useful machine for about $50. We think that $50 (plus a few dollars more for a frame), makes a pretty fair price for creating a functional as well as stylish Linux Toys Digital Picture Frame.

The approach to this project is to:

- Install Red Hat Linux and Toys software on an old laptop computer.
- Add images to the laptop.
- Adapt the laptop computer to fit in a picture frame.
- Display the images in the frame at set intervals.

Because this is primarily a hardware project (the software end is pretty simple), we've included a lot of pictures that Chuck took during the process of building the project. The trick is to get all the hardware out of the laptop case and into the picture frame so that basically all you see is the picture frame (and maybe a little wire or two running off behind it).

When you are done building the Linux Toys Digital Picture Frame, you'll have a way of displaying your family photos that fits into your decor as any other fine accessory would. And if you hate the way you looked at your senior prom, don't worry. The digital picture frame will be on to another shot in a minute.

Figure 15-1 shows what your completed digital picture frame might look like.

Caution We need to start with some serious disclaimers here. First of all, if your laptop has a warranty, you're going to void it. If you're a kid, don't do this without an adult (and especially don't do it to your dad's laptop without asking). You're dealing with electricity here, so unplug the laptop and remove the battery before you start sticking screwdrivers into it. If you can't afford to lose the $50 you paid for the laptop, don't do this project. If you have important data on this machine, for goodness' sake copy it off of there because it's quite possible you will lose it. And by the way, don't run with scissors.

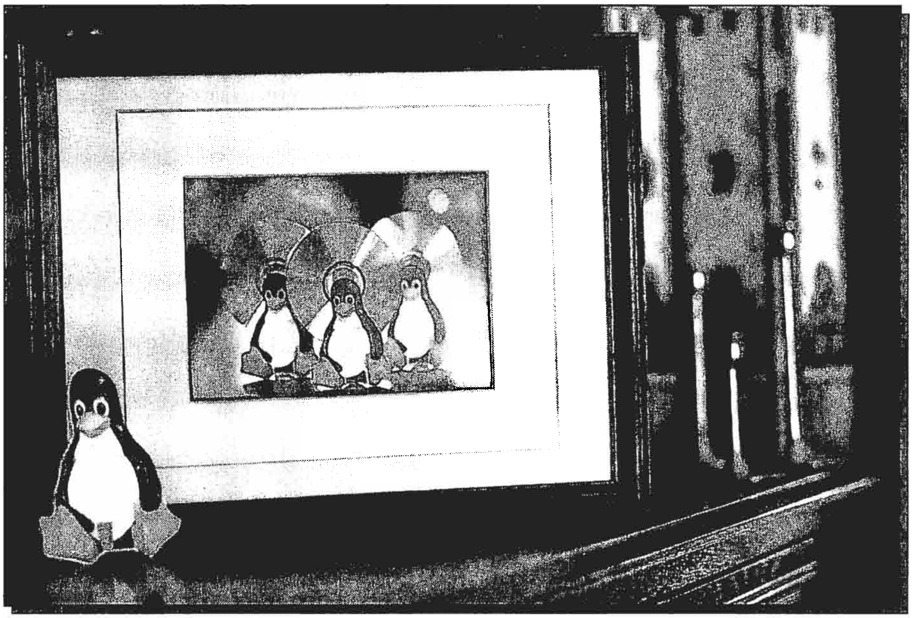


FIGURE 15-1: Cycle through your family photos with the Linux Toys Digital Picture Frame.

Building the Digital Picture Frame

The steps for building your digital picture frame consist of a combination of hardware and software setup. The most important thing to remember is to take your time and try not to break anything.

Step 1: Gather the hardware

The materials you need for this project consist of a laptop PC and some picture-frame materials. You'll need a few tools for the project as well.

The personal computer

As we said, we want to keep the price of the laptop PC down to about $50. We managed to get by with slightly less than the normal Red Hat Linux requirements, partly because we're not running a full-blown Red Hat graphical user interface.

Note

Support for laptops in Linux is generally not as complete as on desktop systems. Many laptops use nonstandard devices for sound cards, graphic chips, and modems. For this project, the key is to have supported graphics chips, a supported processor, enough RAM, and, optionally, a good network card.

If you can, try to find out what the laptop has in it before you buy. Then go to `www.linux-on-laptops.com` to find out if there are any problems using it with Linux.

Most laptops that are compatible with Red Hat Linux should be fine for this project. Check minimums related to CPU and RAM in Appendix C for a Personal Desktop install type. In general, go more for reliability than performance (the laptop doesn't have to do much, but you want it to work for as long as possible).

The laptop that Chuck used to build this project was an old MAG Innovision laptop with the following specifications:

- **CPU**—AMD K6, 333 MHz
- **RAM**—64MB

Figure 15-2 shows a picture of the laptop before we started the project.

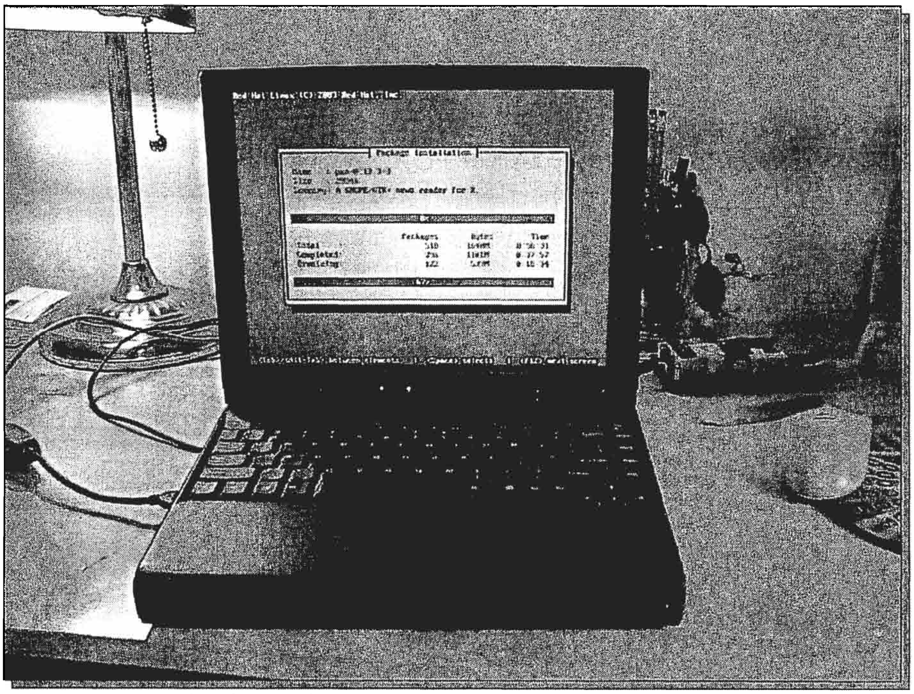


FIGURE 15-2: Get a last look before we take it apart.

Chuck's Tech Tip

Major kudos go to the folks who engineered the MAG Innovision laptops. To say they were engineered for a lifetime of abuse is a bit of an understatement. After getting it open, I almost felt bad taking such a fine machine apart. The same cannot be said for the laptop we tried to use in our first attempt at this project. The laptop was so poorly engineered that it nearly disassembled itself as I was opening it up.

Hard disk

Here is the approximate disk space you will need:

- **Red Hat Linux** — We recommend a Personal Desktop install, which requires about 1.7GB of disk space. A minimum install type will work if you can add the necessary packages (described later) needed to display the images.
- **ltpicframe RPM** — This requires less than 1MB of disk space.
- **Images** — Depending on the size and quality of your images, disk space required to store them will vary. The best idea is to check the size of the digital pictures you are taking. Also, because some older laptops can't do better than a 640 × 480 display, we automatically reduce your images to that size before starting the slide show. So you also have to allow disk space for a 640 × 480 copy of each image. In all, I'd recommend allowing at least 100MB of disk space for your pictures.

CD drive or floppy disk drive

CDs are the preferred medium for installing Red Hat Linux and the Linux Toys software. However, if you are using an old, floppy-only laptop, you can install Red Hat Linux over the network. For that you need a network card (described next).

Network card (optional)

Most laptops built in the past few years have PCMCIA slots built in. Those slots allow credit-card size devices to be inserted into the laptop. By adding an Ethernet card to an old laptop, you can:

- Install Red Hat Linux and Linux Toys without a having a CD drive on your laptop. (You can start the install process from a floppy disk.)
- Use the network connection to load more pictures on the laptop (so you don't need to keep the keyboard attached).

New PCMCIA Ethernet cards go for about $30. Older laptops tended not to have built-in Ethernet; newer ones do. If you don't want to pay for a network card, you're going to need to attach a keyboard and CD drive when you want to add images.

Picture frame

Pieces you need to mount the project are:

- Picture frame
- Matting
- Furring strip (One 8-foot piece about 1/8" × 1.25")

For the picture frame, Chuck bought a frame from a local department store that measures 11" × 14" by about 3" deep. We believe that this should work with most laptops. It cost about $15.

Besides the frame, he bought some matting to cover all but the middle of the screen that displays the picture. When you run the digital picture frame script, you should note the area where the picture appears and get matting that fits around the edges. Figure 15-3 shows an example of the matting we used.

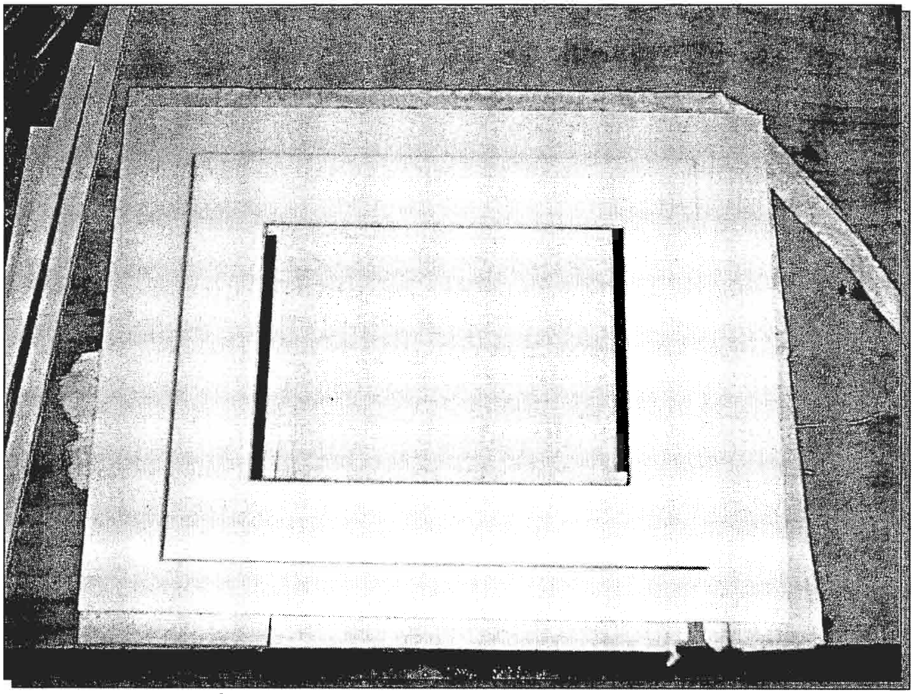


FIGURE 15-3: Matting frames the picture area of the screen.

Tools

You need these tools and accessories to dismantle and install the laptop.

- **Screwdrivers** (regular and Phillips)
- **Tape measure**
- **Anti-static mat**
- **Anti-static bag** (to hold screen temporarily)
- **Waxed paper** (to hold pieces of tape temporarily)
- **Electrical tape**

Step 2: Install and configure Red Hat Linux

Run a Personal Desktop or an Everything install type of Red Hat Linux (see Appendix C). If you want to try to do a smaller installation, you need to make sure that you have at least the following packages installed:

- **XFree86** — Contains the X Window System server that provides the graphical interface needed for the digital picture frame
- **ImageMagick** — Contains the `convert` command, used by the digital picture frame to convert images to the proper size
- **xloadimage** — Contains the `xview` command, used by the digital picture frame to display the images

During configuration of Red Hat Linux, you are asked if you want to configure your network. If you have an Ethernet card on your laptop, I recommend that you configure it to work on your local network. Refer to Chapter 7 for information on configuring a local area network (LAN) and adding a digital picture frame to it.

Step 3: Install ltpicframe

The ltpicframe package comes on the Linux Toys CD. It contains a simple script that takes a directory of images, converts them to appear properly on your screen, and rotates the display of these pictures. To install the ltpicframe RPM, do the following:

1. Log in as root to Red Hat Linux and open a Terminal window (or other shell).
2. Insert the Linux Toys CD and type the following:
   ```
   # mount /mnt/cdrom
   ```
3. Install the ltpicframe package by typing the following:
   ```
   # rpm -Uhv --force /mnt/cdrom/ch15-PictureFrame/ltpicframe*
   ```

Step 4: Get the images

To use the Linux Toys Digital Picture Frame, you need to add images to a directory. You tell the picture frame script the name of that directory, and it will display images from there at set intervals. So right now you want to load up the directory.

By default, we use the `/usr/local/images` directory. You should choose a regular user to own the images directory. That way, you don't have to log in as root to add the images. You should have created at least one regular user account for yourself when you first installed Red Hat Linux. For example, if your user account is named `chuckw`, you should type the following to have that user own the images directory:

```
# chown chuckw /usr/local/images
```

Supported Images

People often save images to JPEG format. However, using the `xview` command, the digital picture frame can display images in other formats as well. You can type the following command to see the graphics types supported by `xview` (a bunch of image types not supported are also displayed but are not listed here):

```
# xview -support
Type Name  Can Dump Description
---------- -------- -----------
niff       Yes      Native Image File Format (NIFF)
jpeg       Yes      JFIF-style JPEG Image
tiff       Yes      TIFF image
pbm        Yes      Portable Bit Map (PBM, PGM, PPM)
```

Loading images

There are lots of different ways to get images onto your laptop for this project. The issue gets a bit trickier once you remove the keyboard and install the laptop into the picture frame. Here are a few suggestions:

Getting images from a CD

If you are running Red Hat Linux from a GNOME or KDE desktop, the contents of any CD you insert are automatically displayed on the desktop. You can open the `/usr/local/images` directory in a file manager window and drag-and-drop the files from the CD directory to that directory.

Getting images from a digital camera

Red Hat Linux includes the gtkam window for downloading images from your digital camera. Here's a quick procedure for downloading images from your digital camera in Linux:

1. Connect your digital camera to the USB or serial port on your laptop (assuming it has one). Make sure that the camera is on and set to download.
2. From the Red Hat menu, click Graphics ➔ Digital Camera Tool. The gtkam window should appear.
3. From the gtkam window, select Camera ➔ Add Camera. A pop-up window appears.
4. Click Detect. Gtkam will try to determine what camera you are using. If the camera is properly detected and the right port is selected, the name of the camera and port it is connected to should appear in the window.
5. Click Apply and OK.
6. In the left column, select your digital camera, and then select the directory containing the images. Thumbnails should be downloaded on the right side of the window, as shown in Figure 15-4.

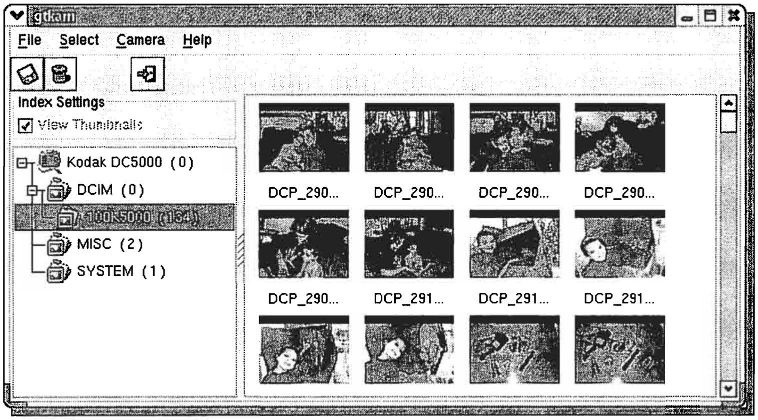


FIGURE 15-4: Use gtkam to download images from a digital camera.

7. Click the images you want to download (or click Select ➔ All for all images).
8. Click File ➔ Save Selected Photos.
9. When the pop-up window appears, choose `/usr/local/images` as the download directory. You can use the filenames given by the camera or attach your own filename to each photo.

If your images need any modification after the download, you can use The Gimp (select Graphics ➔ The GIMP) to modify them. In particular, you may want to rotate, crop, or resize the images.

Getting images from the network

If you have an Ethernet connection from your laptop to your own LAN, you can copy files to your laptop over that connection. Using the user login that you assigned to the `/usr/local/images` directory, here are a few ways you can install the images on your laptop.

- **Remote copy** — With the SSH service turned on (which it is by default), you can use the `scp` command to copy files from another Red Hat Linux computer on your LAN. Here's an example of the `scp` command:

  ```
  $ scp * chuckw@picframe:/usr/local/images
  ```

  In this example, all files from the current directory (assumed to be full of images) are copied to the `/usr/local/images` directory on the computer named `picframe` (assumed to be your laptop). Replace `chuckw` with your user name on the digital picture frame and type the password when you are prompted.

- **Remote ssh (and X)** — A very cool feature of `ssh` is that, by default, when you log in to Red Hat Linux using `ssh`, any GUI window you launch from the remote system will appear on your local desktop. This can be especially useful after the laptop has been put in the frame.

So, for example, with Red Hat Linux running on a computer named `toys`, you could log in (as `chuckw`, for example) to your picture frame laptop (called `picframe`, for example) as shown here:

```
$ ssh chuckw@picframe
```

Type your password as prompted and you will see a shell prompt. Now, if you still have a serial or USB port on your laptop, plug in your camera and type:

```
$ gtkam &
```

Now the gtkam window you started from `picframe` displays on your desktop on `toys`. You can operate this, or any GUI window, without having a mouse or keyboard on the picture frame computer.

- **FTP** — You can configure your digital picture frame as an FTP server (see Chapter 11). Then you could use any FTP client program to transfer files to the digital picture frame. (FTP clients are readily available for Windows, Macintosh, Linux, and other computers that may not have `ssh` clients.)

Step 5: Configure the picture frame software

You're going to set up the digital picture frame software to run automatically when you start up the laptop. The advantage of this is that you have only to turn on the computer and it will begin displaying your pictures. Here is what you do:

Caution

Before you edit the `/etc/inittab` file, make a backup copy of it and have a boot disk available (just in case). A mistake in the `inittab` file can result in an operating system that will not boot.

1. From a Terminal window (or other shell), open the `/etc/inittab` file as root user using any text editor.
2. Find the `initdefault` line and change it to run level 4. It should appear like this:

   ```
   id:4:initdefault
   ```

3. Add the following lines to the bottom of the file:

   ```
   pfin:4:respawn:/usr/local/bin/ltPF_X.sh
   ```

4. Restart the computer. The digital picture will start up. Here's what it does:
   - The `xview` command displays an image from the `/usr/local/images` directory.
   - After five seconds, the next picture is displayed.

   If you want to change either of those values, open the `/usr/local/bin/ltPF-X.sh` file in any text editor. In that file, change either the value of IMAGE_PATH or SLIDE_DELAY to change the location of the images directory or the number of seconds to display each picture before changing it, respectively.

At this point, you have a working digital picture frame. You should be able to switch it on and have the pictures begin displaying after a minute or two. Next, let's hope, you'll be able to keep it working after you take it apart to put it in a picture frame.

Step 6: Tune your laptop

You want your digital picture frame to run continuously and not shut down. To make sure that your screen doesn't blank out after a while, you should go into the computer's BIOS and disable power management. Power-management features take the form of turning off the screen, shutting down the hard drive, suspending, and processing. Disable all that if you can.

Step 7: Adapt the laptop to the picture frame

With the laptop now set up to play the pictures, you can start adapting it to fit in a picture frame. Besides the laptop, you need a picture frame and some tools (described in Step 1). In the process of adapting the laptop we used, we removed the following components:

- CD-ROM drive
- Floppy drive
- Keyboard
- Battery
- Modem
- Lots of plastic

Chuck also wanted to remove the mouse pad, but it was used for connecting the hard drive and the BIOS battery, so he couldn't. You might be able to remove yours, however.

After removing components, you'll need to have the keyboard handy (or have an extra keyboard to plug into the PS/2 port on the back), so you can turn off the components in the BIOS that you remove. Basically, what you will have left are:

- Motherboard
- Screen
- Fan
- Optional network card (in PCMCIA slot)

With the approach we took, you will be able to connect to the digital picture frame through your Ethernet card without opening the frame. To use the keyboard or other components, you'll have to open the frame and reconnect them.

Chuck's Tech Tip

For what I did, you don't need to do any soldering. If you want to get fancy and have all of the ports hanging out the back, you'll have to remove the ports from the motherboard. You can do that with a soldering iron and some wire (probably just some 24-gauge wire that comes in a Category 5E cable; as long as it's less than six or so inches) to extend the port.

Every laptop is different. So the step-by-step example that we include here may not even apply to your laptop. There are a few tips, however, that you should keep in mind no matter what laptop you are using:

- After you have unscrewed a component, slowly and gently remove it! When you move too quickly, it's easy to break off a cable or drop screws or other pieces into the laptop.
- LCD screens are very fragile. Pressing too hard on the screen or dropping it will likely damage it to the point where you can't use it.
- Keep all the parts as you dismantle the laptop. You may need some of them later as you mount it in the picture frame. Or you may decide to make it back into a laptop later. Static shields can be reused to prevent short circuits.
- Use a lot of electrical tape to make sure you don't get a short and destroy the hardware. Don't use electrical tape for anything structural or on the matting, however, since it contains oil that can bleed into the matting. Chuck used waxed paper to keep LCD wiring tape from sticking to anything.
- Boot up your laptop a few times along the way. No use in mounting a digital picture frame that is broken.

Here are the steps for dismantling a laptop for this project, based on what Chuck went through to modify his MAG Innovision laptop into a Linux Toys Digital Picture Frame:

1. Check that the digital picture frame software is installed and running properly on the laptop before beginning. Then power down and unplug the computer.
2. Remove the battery.

Caution Be sure there is no power to the laptop before you proceed.

3. Remove the screws holding the LCD screen to the case (or, if the case is held together by tabs, gently pry apart the tabs). Screws may be covered by tape or rubber plugs on the front of the screen frame.
4. Gently separate the LCD screen, as shown in Figure 15-5. Be careful:
   - Don't break the small latches that attach the frame around the screen.
   - Don't break the cables connecting to the LCD.
   - Don't press on the screen or hit it with any sharp tools.
5. Gently fold the screen down and identify the high-voltage wires and the data port ribbon cable.
6. Disconnect the high-voltage wires and data-port ribbon cable; then remove the screen from the laptop. (Store the screen in an anti-static bag. The screen may be the most expensive part to replace, so treat it kindly.)

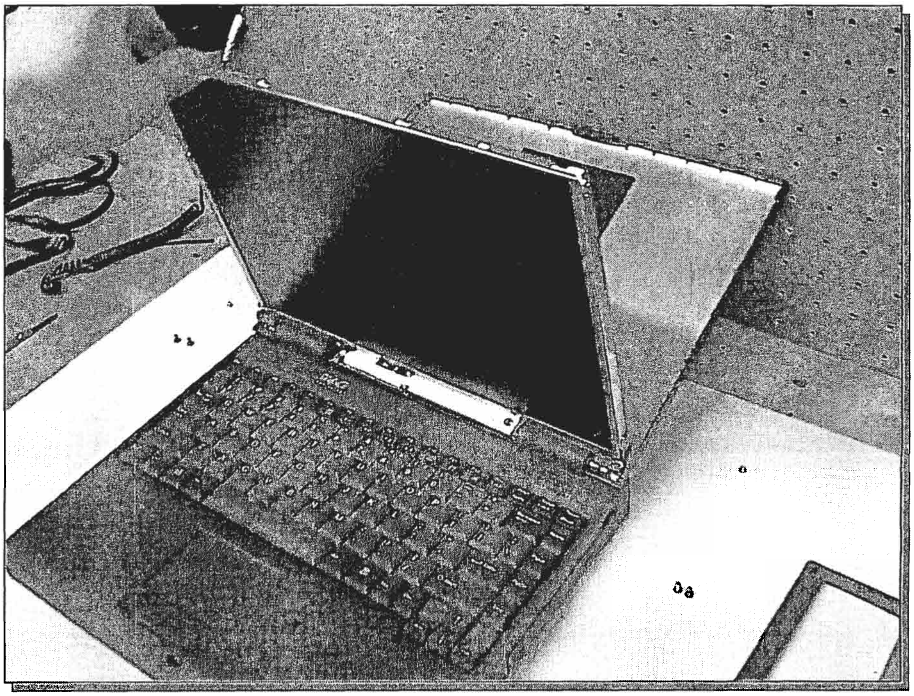


FIGURE 15-5: Carefully separate the LCD screen from the case.

7. Remove the plastic cover over the mouse-pad area (possibly from screws underneath the laptop).
8. Remove the hinges and plastic at the back of the laptop. At this point, the laptop should look similar to the one shown in Figure 15-6.
9. Remove the keyboard and carefully unplug it.
10. In our example, the laptop had a metal case surrounding most of the components. If your laptop has such a case, remove it.
11. Now it's time to just start unscrewing stuff and removing components. The order in which you remove components depends mostly on what is in the way. You will remove some components temporarily, just to extract everything from the case.
12. Remove all remaining plastic so that you can separate the motherboard from what is left of the case.

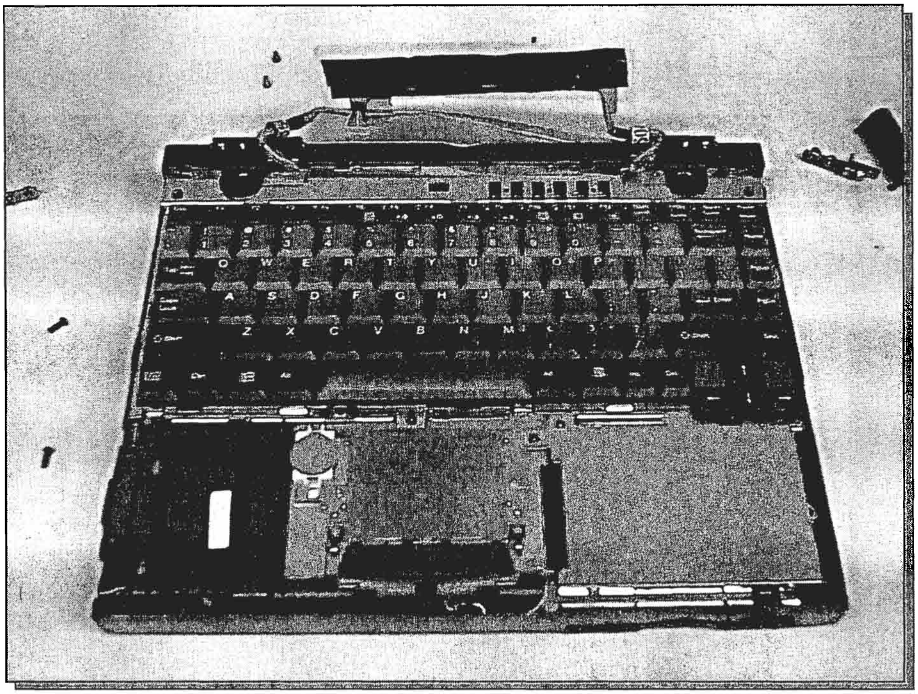


FIGURE 15-6: Remove plastic casing over mouse pad and around the screen.

13. Once the motherboard is removed, place it on the anti-static mat and reattach the components necessary to have the following setup:
    - Screen
    - Cord to power supply
    - Hard disk
    - Mouse pad. (We needed this because power went through it to the hard disk. You may not need it.)
    - Keyboard. (We attached an external keyboard to the PS/2 port temporarily.)
    - PCMCIA card. (This is optional. We inserted an Ethernet card and added a dongle to connect the digital picture frame to the LAN. A dongle is a wire with a connector on the end, in this case, to connect an RJ-45 LAN cable.)

    Figure 15-7 shows what the project might look like at this point. Figure 15-8 shows the underside of the motherboard, with the PCMCIA network card.

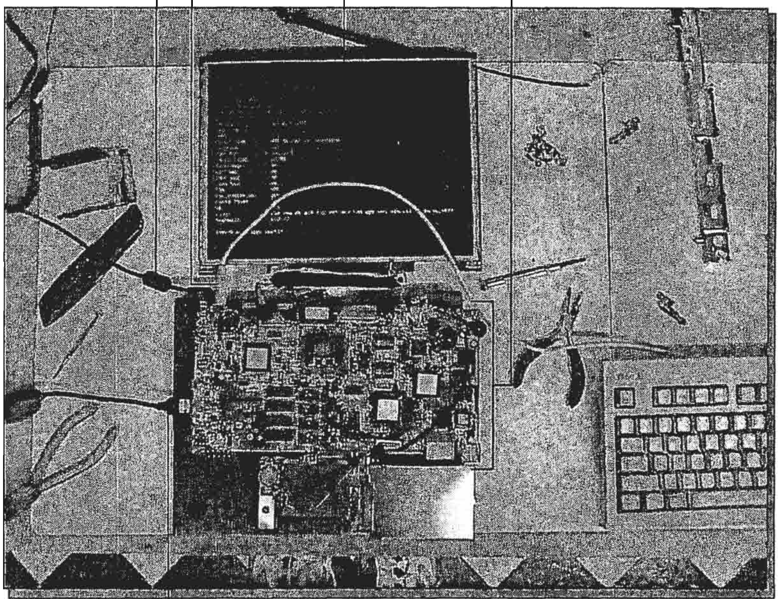


FIGURE 15-7: Get it down to the motherboard, screen, keyboard, and optional network card.

14. Boot up the laptop and go into setup mode before it boots (usually the F2 function key).
15. In setup mode, disable the CD-ROM drive and floppy drive. Save the changes and continue to boot to the picture frame.
16. If the pictures begin to play, measure the area on the screen where you can see the pictures. Then cut the picture frame matting so that only that area will appear through the matting. Chuck used two layers of matting.

Note

Because the laptop should now boot immediately to the picture frame, if you want to get to the operating system for some reason, you can press Ctrl+Alt+F2 to get to a virtual terminal and login. To go back to where pictures are being displayed, type exit; then press Ctrl+Alt+F7.

17. Power down the laptop, remove the keyboard and screen, and disconnect the power supply.

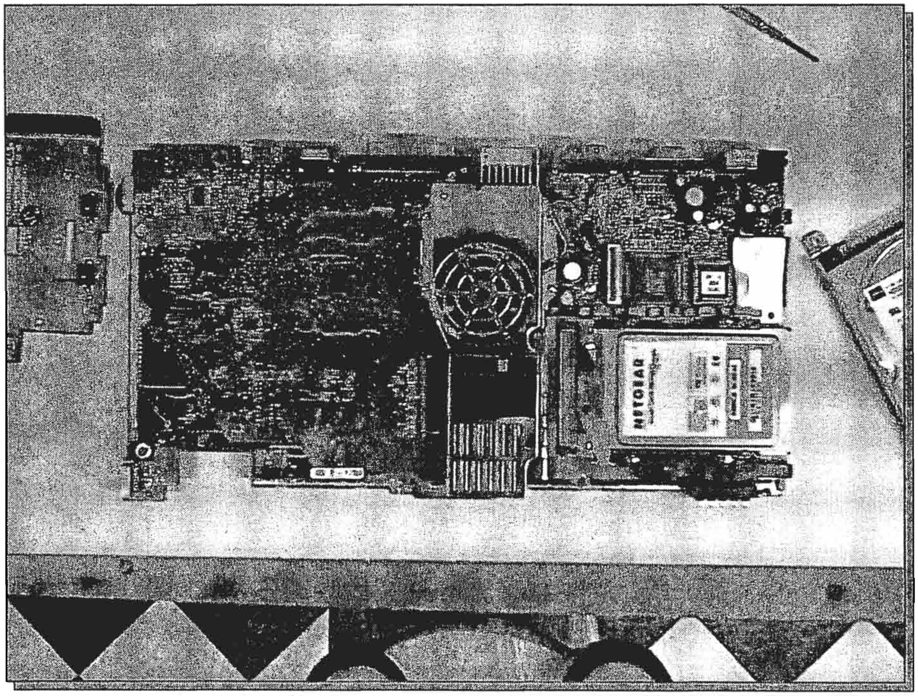


FIGURE 15-8: On the underside of the laptop, you can see the network card.

Chuck's Tech Tip

You'll want to ensure that you aren't going to have a short anywhere. Since you're essentially repackaging a carefully engineered product, you have to remember to use electrical tape, or a similar insulator, to cover exposed metal where it will touch other gear (such as the power supply or the back of the LCD screen).

18. Place the matting inside the picture frame.
19. Position the LCD screen so that the viewable area of the picture frame shows through the opening.

Note

Be careful when mounting the LCD. It has sensitive components, and too much pressure could break it. Mount it firmly, but not so tightly that you risk breaking it.

20. Use small sections of furring strip on all four sides to position the LCD screen in the frame.
21. Use two lengths of furring strip and some wood screws to run across the back of the LCD to hold it all in place. Figure 15-9 shows the back of the picture frame, with furring strips holding the screen in place.

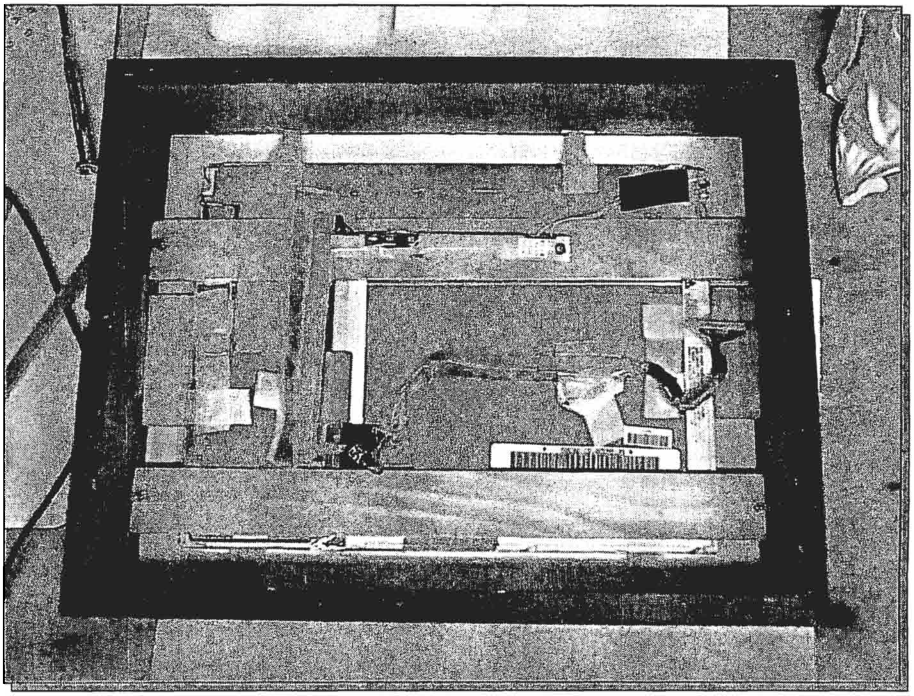


FIGURE 15-9: We used furring strips to hold the screen in place.

22. Reconnect the power and screen cables between the motherboard and the screen. Figure 15-10 shows an example.
23. Position the motherboard, fan, and hard disk in the case.
24. Connect the fan, power supply, dongle to the Ethernet card (if the card is included), and the power supply to the digital picture frame. Figure 15-11 shows an example of the final components in place.
25. Use two more furring strips across the back to hold the motherboard, hard disk, and the power supply in place.
26. Notch out places for the power cable and (optionally) the network card. Chuck used a zip tie to keep the cables nice and neat.
27. Cut a ventilation hole about the same size as the one on the laptop. Preferably, the hole should be over the fan to promote airflow. Install the back (securing it in place).
28. Turn on the picture frame and see if it works. If it does, you should see something like the frame in Figure 15-1.

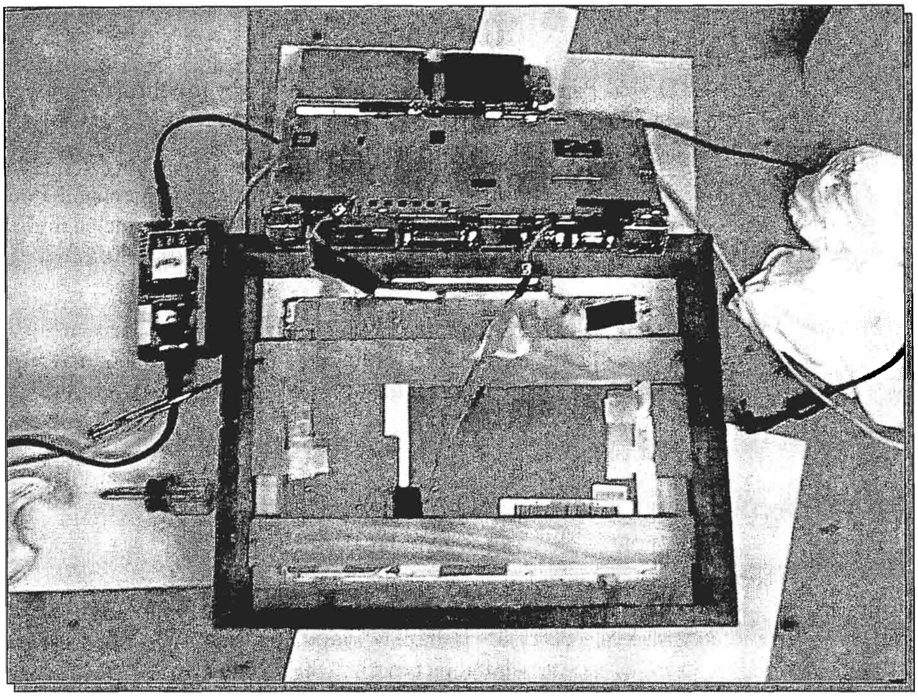


FIGURE 15-10: Connect the power and screen cables from motherboard to LCD.

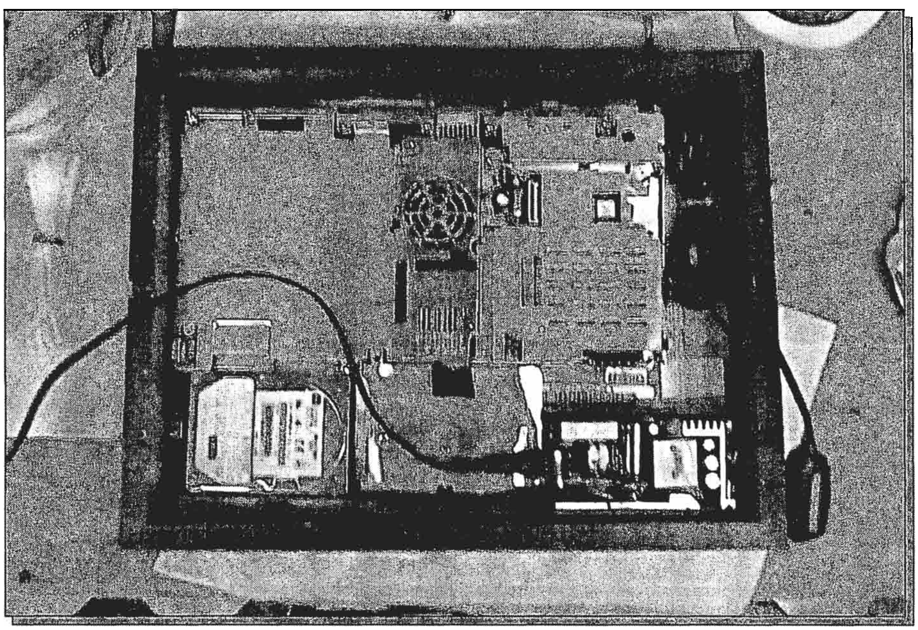


FIGURE 15-11: Place the motherboard and the final components in the frame.

Using Your Digital Picture Frame

The Linux Toys Digital Picture Frame was designed to run by itself. At this point, you don't have to do anything but turn it on and off. However, by either reattaching a keyboard or using the network card, you can add and delete images for the digital picture frame.

We didn't make serial, USB, and ports other than the Ethernet card accessible from the digital picture frame. If you made those ports available, however, you could attach a digital camera or other device to the digital picture frame, then access the laptop via the Ethernet card (as described in the "Loading Images" section earlier in this chapter).

To turn off the digital picture frame, you can usually just unplug it. Not too many files are open, so you should be okay. A cleaner way to shut down would be to log in as root user (using the `ssh` command over your Ethernet connection). Then type shutdown to turn off the digital picture frame before unplugging it.

Summary

An old laptop can serve well as an attractive digital picture frame. If you can carefully remove the laptop components from the case, then reassemble the pieces you need in a picture frame, you can end up with a nice accessory to display your digital images. With an optional network card added, you can access your digital picture frame from your LAN to periodically update your catalog of images.

appendix A

in this appendix

☑ Getting Red Hat Linux software

☑ Using the Linux Toys CD

☑ Understanding the Linux Toys packages

Getting the Software

All of the software used for the Linux Toys project comes from one of two places: Red Hat Linux or the Linux Toys CD. The first you have to get yourself; the second comes with this book.

Red Hat Linux

The complete Red Hat Linux installation comes on three CDs. For our projects, we are using the i386 (in other words, the PC) version of Red Hat Linux. Here are a couple of ways to get Red Hat Linux:

- For the full set of Red Hat Linux installation CDs, along with instructions on how to use and administer Red Hat Linux, you can purchase *Red Hat Linux Bible* (written by Christopher Negus and published by Wiley). The content of these CDs is the same as the content available for download from Red Hat's Web site.
- You can download images of the installation CDs from many different Red Hat Linux mirror sites. You can then burn those CD images onto CDs or install the software from those images from hard disk or over a network.

Linux Toys projects were tested on Red Hat Linux 9, but presumably they should work on later versions of Red Hat Linux. Check the `LinuxToys.net` site for information on any incompatibilities that may arise from later versions of Red Hat Linux.

Cross-Reference

Installation of Red Hat Linux is described in Appendix C.

The Linux Toys CD

The Linux Toys CD that comes with this book contains software packages in Red Hat Package Management (RPM) format, tar/bzip2 format, and even a few floppy disk images.

Refer to each project chapter for information on which Linux Toys packages are needed. Most often, you can install the Linux Toys package required by mounting the CD, then using the included `install.me` command, as follows:

```
# mount /mnt/cdrom
# /mnt/cdrom/ch03-ltJukebox/install.me
```

Change *ch03-ltJukebox* to the chapter you are trying to install. Occasionally, there are some extra installation instructions. These are described in detail in their respective chapters.

Linux Toys Packages

Chapter 3 (/mnt/cdrom/ch03-ltJukebox):

- **install.me:** Project installation script
- **COPYRIGHT:** Copyright information for included music
- **Listen.ogg:** Music by Bill Broomall and Richard Williams
- **On_the_Porch.ogg:** Music by Bill Broomall
- **ltJukebox-1.2-3.noarch.rpm:** Primary project package file
- **cddbd-1.4-0.i386.rpm:** CD database daemon package
- **freedb-complete-20030701.tar.bz2:** CD information database
- **perl-MP3-Info-1.01-0.i386.rpm:** CD naming PERL library

Chapter 4 (/mnt/cdrom/ch04-VideoArchive):

- **install.me:** Project installation script
- **a52dec-0.7.4-0.i386.rpm:** Library for decoding AC-3 streams
- **aalib-1.4rc5-fr2.i386.rpm:** ASCII art low level library
- **alsa-lib-0.9.4-fr2.i386.rpm:** Advanced Linux Sound Architecture
- **avi2vcd-1.0-0.noarch.rpm:** Linux Toys AVI2VCD conversion script
- **avifile-0.7.35-1.i386.rpm:** AVI file library
- **avifile-codecs-20020516-1.i386.rpm:** Binary win32 video codecs
- **divx4linux-5.01-0.dag.rh73.i386.rpm:** DivX Codec 5.01
- **faad2-1.1-fr2.20030409.i386.rpm:** Decodes MPEG2/4 AAC

- **ffmpeg-0.4.6-0.i386.rpm:** Audio/Video Swiss army knife
- **libdv-0.99-fr2.i386.rpm:** IEEE 1394 video codec
- **libdvdread-0.9.4-fr3.i386.rpm:** Foundation for reading DVD disks
- **libpostproc-0.90-fr2.i386.rpm:** mplayer postprocessing component
- **libquicktime-0.9.0-1.i386.rpm:** QuickTime manipulation library
- **lirc-0.6.6-fr1.i386.rpm:** Linux infrared remote control package
- **lzo-1.08-0.i386.rpm:** Lossless data compression
- **mjpegtools-1.6.1-1.i386.rpm:** MPEG audio and video tools
- **mplayer-0.90-fr2.i386.rpm:** The best movie player for Linux
- **mplayer-fonts-1.0-fr1.noarch.rpm:** Font files for mplayer
- **nvrec-20030316-3.i386.rpm:** Video capture core
- **vcdimager-0.6.2-1.i386.rpm:** VCD (pre-)mastering and ripping tool
- **xvidcore-0.9.1-fr2.i386.rpm:** Open implementation of the DivX codec

Chapter 5 (/mnt/cdrom/ch05-WebVCR):

- **install.me:** Project installation script
- **aalib-1.4rc5-fr2.i386.rpm:** ASCII art low-level library
- **alsa-lib-0.9.3-fr2.i386.rpm:** Advanced Linux Sound Architecture
- **avifile-0.7.35-1.i386.rpm:** AVI file library
- **avifile-codecs-20020516-1.i386.rpm:** Binary win32 video codecs
- **divx4linux-5.01-0.dag.rh73.i386.rpm:** DivX Codec 5.01
- **faad2-1.1-fr2.20030409.i386.rpm:** Decodes MPEG2/4 AAC
- **ffmpeg-0.4.6-0.i386.rpm:** Audio/Video Swiss army knife
- **libdv-0.99-fr2.i386.rpm:** IEEE 1394 video codec
- **libdvdread-0.9.4-fr3.i386.rpm:** Foundation for reading DVD disks
- **libpostproc-0.90-fr2.i386.rpm:** mplayer postprocessing component
- **libquicktime-0.9.0-1.i386.rpm:** QuickTime manipulation library
- **lirc-0.6.6-fr1.i386.rpm:** Linux infrared remote control package
- **lzo-1.08-0.i386.rpm:** Lossless data compression
- **mplayer-0.90-fr2.i386.rpm:** The best movie player for Linux
- **mplayer-fonts-1.0-fr1.noarch.rpm:** Font files for mplayer
- **nvrec-20030316-3.i386.rpm:** Video capture core

- **perl-Class-MethodMaker-1.10-8.i386.rpm:** MethodMaker module
- **perl-Compress-Zlib-1.22-8.i386.rpm:** Zlib module
- **perl-DateManip-5.42-0.9.noarch.rpm**: Date manipulation module
- **perl-HTML-Parser-3.26-17.i386.rpm:** Perl HTML parser library
- **perl-HTML-TableExtract-1.08-8.i386.rpm:** TableExtract module
- **perl-HTML-Tree-3.17-8.i386.rpm:** HTML Tree module
- **perl-Lingua-EN-Numbers-Ordinate-0.01-8.i386.rpm:** PERL module
- **perl-Lingua-Preferred-0.2.2-8.i386.rpm:** Lingua Preferred module
- **perl-Log-TraceMessages-1.3-8.i386.rpm:** Log tracing module
- **perl-Term-ProgressBar-2.03-8.i386.rpm:** Status indicator module
- **perl-TermReadKey-2.20-7.i386.rpm:** Terminal key access module
- **perl-Tie-IxHash-1.21-8.i386.rpm**: Tie-IxHash module
- **perl-Tk-800.024-2.i386.rpm:** Perl Tk module
- **perl-Tk-TableMatrix-1.01-8.i386.rpm:** TableMatrix module
- **perl-Tk-JPEG-2.014-8.i386.rpm:** JPEG module
- **perl-Unicode-String-2.07-8.i386.rpm:** Unicode Perl module
- **perl-XML-DOM-1.42-8.i386.rpm:** XML-DOM module
- **perl-XML-Simple-2.08-8.i386.rpm:** Basic XML manipulation
- **perl-xmltv-0.5.15-1.i386.rpm:** TV listings utility
- **perl-XML-Twig-3.09-3.noarch.rpm:** XML-Twig Perl module
- **perl-XML-Writer-0.4-8.i386.rpm:** XML-Writer Perl module
- **xmltv-0.5.15-29.noarch.rpm**: Xmltv commands for grabbing TV listings.
- **xvidcore-0.9.1-fr2.i386.rpm:** Open implementation of the DivX codec
- **webvcrplus-0.9.2.8-1.noarch.rpm:** WebVCRplus package

Chapter 6 (/mnt/cdrom/ch06-Arcade):

- **install.me:** Project installation script
- **alsa-lib-0.9.4-fr1.i386.rpm:** Advanced Linux Sound Architecture
- **grustibus-0.43.CVS20020410-3.i386.rpm:** XMAME Frontend
- **xmame-0.67.2-fr0.rh73.1.i386.rpm:** X MultiArcade Machine Emulator
- **xmame-roms-0.67.2-fr0.rh73.1.i386.rpm:** Freely available ROMS
- **xmame-x11-0.67.2-fr0.rh73.1.i386.rpm:** XMAME for X11 DGA/XV

Chapter 8 (/mnt/cdrom/ch08-HomeBroadcast):

- **install.me:** Project installation script
- **a52dec-0.7.4-0.i386.rpm:** Library for decoding AC-3 streams
- **aalib-1.4rc5-fr2.i386.rpm:** ASCII art low level library
- **alsa-lib-0.9.3-fr2.i386.rpm:** Advanced Linux Sound Architecture
- **avifile-0.7.35-1.i386.rpm:** AVI file library
- **avifile-codecs-20020516-1.i386.rpm:** Binary win32 video codecs
- **divx4linux-5.01-0.dag.rh73.i386.rpm:** DivX Codec 5.01
- **faad2-1.1-fr2.20030409.i386.rpm:** Decodes MPEG2/4 AAC
- **ffmpeg-0.4.6-0.i386.rpm:** Decodes MPEG2/4 AAC
- **libdv-0.99-fr2.i386.rpm:** IEEE 1394 video codec
- **libdvdread-0.9.4-fr3.i386.rpm:** Foundation for reading DVD disks
- **libpostproc-0.90-fr2.i386.rpm:** mplayer postprocessing component
- **libquicktime-0.9.0-1.i386.rpm:** QuickTime manipulation library
- **lirc-0.6.6-fr1.i386.rpm:** Linux infrared remote control package
- **lzo-1.08-0.i386.rpm:** Lossless data compression
- **mplayer-0.90-fr2.i386.rpm:** The best movie player for Linux
- **mplayer-fonts-1.0-fr1.noarch.rpm:** Font files for mplayer
- **xvidcore-0.9.1-fr2.i386.rpm:** Open implementation of the DivX codec

Chapter 9 (/mnt/cdrom/ch09-DigiTemp):

- **install.me:** Project installation script
- **ltweather-1.0-1.noarch.rpm:** DigiTemp software plus add-ons

Chapter 10 (/mnt/cdrom/ch10-DigitalReceptionist):

- **install.me:** Project installation script
- **ltdrconfig-1.1-0.noarch.rpm:** Digital receptionist configuration script
- **mgetty-vocp-1.1.30-3.i386.rpm:** mgetty patched with VOCP patches
- **normalize-0.67.6-0.i386.rpm**: Commands to normalize audio
- **perl-Audio-DSP-0.02b-8.i386.rpm:** Audio-DSP Perl module
- **perl-CGI.pm-2.97-8.i386.rpm:** CGI.pm Perl module
- **perl-Crypt-Blowfish-2.09-8.i386.rpm:** Crypt-Blowfish Perl module

- **perl-Crypt-CBC-2.08-8.i386.rpm:** Crypt-CBC Perl module
- **perl-Modem-Vgetty-0.04-8.noarch.rpm:** Modem-Vgetty Perl module
- **perl-Tk-800.024-8.i386.rpm:** Tk Perl module
- **perl-Tk-JPEG-2.014-8.i386.rpm:** Tk-JPEG Perl module
- **perl-VOCP-0.9.3-8.noarch.rpm:** VOCP Perl module
- **perl-XML-Mini-1.2.8-8.i386.rpm:** XML-Mini Perl module
- **vocp-0.9.3-0.i386.rpm:** Full-featured telephone messaging system

Chapter 13 (/mnt/cdrom/ch13-DogHouse):

- **install.me:** Project installation script
- **doghouse.img:** Floppy disk image of DogHouse Linux
- **rawrite.exe:** DOS/Windows disk imaging program
- **bsd-games-2.9-9.i386.rpm:** BSD games package

Chapter 14 (/mnt/cdrom/ch14-RemoteControl):

- **install.me:** Project installation script
- **ltrc-1.3-1.noarch.rpm:** Remote control scripts
- **perl-TermReadKey-2.20-7.i386.rpm:** ReadKey Perl module
- **perl-Time-HiRes-1.38-3.i386.rpm:** High resolution timer Perl module

Chapter 15 (/mnt/cdrom/ch15-PictureFrame):

- **install.me:** Project installation script
- **ltpicframe-1.2-1.noarch.rpm:** Picture frame slideshow script

Source Code (/mnt/cdrom/SOURCE):

- Contains all source code for Linux Toys projects, including code that can be used to try building Linux Toys on versions of Linux other than Red Hat

appendix B

in this appendix

- ☑ Using the shell or graphical interfaces
- ☑ Linux files and directories
- ☑ The root user account and file permissions
- ☑ How running processes work
- ☑ Editing text files
- ☑ How to find more information

ABCs of Using Linux

You don't need to know a lot of Linux to build these projects . . . but you do have to know a few things. If you are spanking, brand new to Linux, I strongly recommend that you go through this appendix. It will help you:

- Get logged in.
- Get around.
- Get permission.
- Get working.

The last bit of help this appendix will get you is information about where to get more help. In particular, *Red Hat Linux Bible* (also by Christopher Negus) can give you the background you need to expand your use of Linux in general. (It also contains the Red Hat Linux CDs that you need with this book.)

The Shell and the GUI

There are two basic ways to use Linux:

- **The shell** — In the beginning, there were no colors, icons, or windows. If you wanted the computer to do something, you had to type a command using what was referred to as the shell command line interpreter (or simply the shell).
- **The GUI** — To make computers more intuitive to use, the graphical user interface (or GUI) was developed. Instead of making you type commands to make the computer do things, the GUI allowed you to use a mouse (or other type of pointing device) to start programs, open files, and generally work with your computer.

Linux offers both shell and GUI methods for operating your computer. For Linux Toys, there are times when I'll ask you specifically to use either the GUI or the shell (usually because the particular tool I want you to use is either a GUI window or shell command). Therefore, quick primers in GUIs and shells are in order.

The GUI

GUI stands for graphical user interface. If you selected to have a graphical login when you installed Red Hat Linux, Linux will automatically start up your computer session in GUI mode after you log in. This means you will be able to operate your computer just as you do with a Mac or Windows computer, using:

- A mouse
- A keyboard
- A graphical screen (with windows, icons, panels, and menus)

For Linux, there is one major facility providing the foundation for all graphical interfaces (the X Window System), two major desktop environments to use with X (GNOME or KDE), dozens of window managers to provide the look and feel of the desktop (metacity, windowmaker, twm, and so on) and thousands of ways you can customize your own desktop. Now, forget I said all that.

To simplify using Red Hat Linux, Red Hat gives you two basic choices of GUI: GNOME or KDE. Red Hat made the two environments look pretty much the same. So, whether you are using GNOME or KDE in Red Hat Linux, you will start out with the same:

- Background
- Panel buttons
- Desktop icons
- Main menu (pretty much)

I've chosen to describe the GNOME desktop because that's what Red Hat uses as its default. Figure B-1 shows an example of the GNOME desktop.

Touring the GUI

I opened two windows in the previous figure. The one at the top is a Terminal window, which opens a shell for typing commands. The lower window contains icons for changing preferences of how your desktop works and appears. The panel runs along the bottom of the screen and a few icons appear on the desktop. Instead of describing all that stuff in detail, I want to start by taking you on a little tour.

1. **Log in** as a regular user.
2. **Start the GUI:** If the GUI appears (as shown in Figure B-1), proceed to the next step. Otherwise, try typing:

   ```
   startx
   ```

   This should start your desktop. If `startx` fails saying `startx: command not found` the desktop software may not be installed (see Appendix C for how to install additional Red Hat software, GNOME in this case). If the `startx` command is found but still fails, run the `redhat-config-xfree86` command to configure your video card. There are a few tips on setting up a video card in Chapter 4. If you still have problems, check out `www.xfree86.org` to try to figure out why your GUI isn't working.

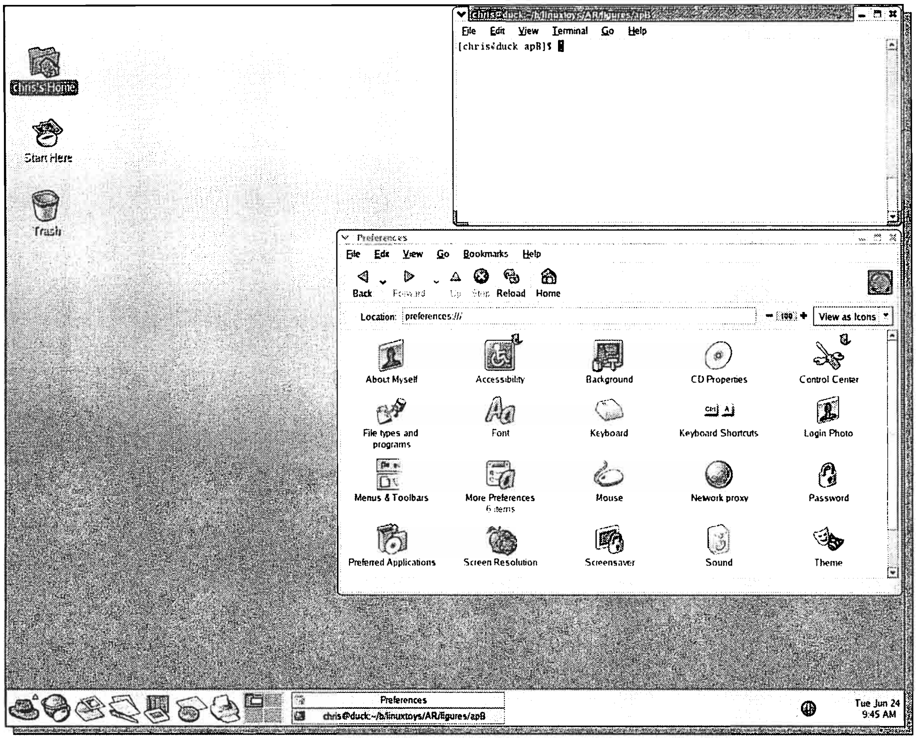


FIGURE B-1: The GNOME desktop offers simple controls and menus.

3. **Run the File Manager:** With the GNOME desktop appearing on the screen, double-click the Home directory icon (it's named something like `chris's Home` depending on your user name). This will open a file manager window, which for GNOME is the Nautilus window. Figure B-2 shows an example of Nautilus.

   Here are a few things to check out about the Nautilus file manager window:

   - **Home Directory:** Your home directory is, by default, your user name in `/home`. So, for the user name `chris`, the home directory would be `/home/chris`. Your home directory is where you store your personal data and applications. Create as many folders and subfolders in there that you want.
   - **Getting Around:** Double-click a folder to move down the directory structure. Click the Up button to move up. Or you can type a directory into the Location box and press Enter to open a particular directory or file.
   - **Files & Applications:** Double-click a file or application to launch it. Right-click a file, and then select Open With to choose an application that can be used to open the file. Some file types (such as image files) can be opened with different applications (such as image viewers, Web browsers, or manipulation tools.)

   Close the Nautilus file manager window by clicking File ➔ Close Window.

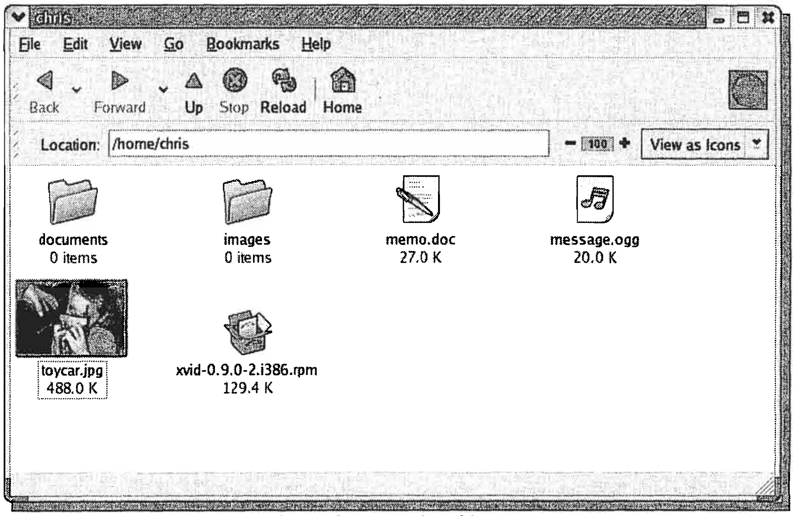


FIGURE B-2: Navigate Linux from the Nautilus file manager.

4. **Open the Red Hat Menu:** Click the red hat that appears on the panel at the bottom of the screen. From the menu that pops up, select from categories of application programs to run. Try out any of the applications that interest you. In terms of getting your Linux system set up the way you like, pay particular attention to these submenus (you'll need the root password to use many of these features):
   - **Preferences:** Change how your desktop looks and behaves from selections on this menu. I usually change CD Properties (to turn off automatic CD detection), Screensaver (to choose a screen saver), and Background (to select an image for my background).
   - **System Settings:** Configure your date/time, display, keyboard, network, and other basic system services.
   - **System Tools:** Check the status of your hardware, system logs, printer, and system tasks.
5. **Check out the Panel:** Move your cursor across the Panel at the bottom of the screen. Tool Tips tell you which application is launched for each icon. Click the Workspace switcher to switch among four different workspaces (this gives you four times the screen real estate to work on). Right-click in the panel; then click Add to Panel to select an applet (small application) to run on the panel.

Checking out Red Hat configuration tools

Over the past few years, Red Hat has been developing its own graphical administrative tools for configuring Red Hat Linux. Lately, these tools have matured to the point where they are actually the best way of administering most basic Linux services. I figure that Red Hat must

feel good about these windows, since they removed Terminal icons from the desktop (assuming that someone who doesn't already know how to get to a shell doesn't need to).

Table B-1 lists the Red Hat configuration windows, along with how to launch them from the GUI or from a Terminal window.

Note

While most of the Red Hat configuration tools are meant to run on a GUI, some have text modes as well. This can be useful if, for example, your video card isn't working and you need to run redhat-config-xfree86 from your shell to get the GUI working at all.

Table B-1 Red Hat Configuration Windows

Window Name	*From Red Hat Menu, click:*	*From Terminal window, type:*	*Descriptions*
Add/Remove Applications	System Settings➔Add/Remove Applications	redhat-config-packages	Work with Red Hat software packages.
Authentication Configuration	System Settings➔Authentication	authconfig-gtk	Change how users are authenticated.
CD Writer	System Tools➔CD Writer	xcdroast	Create your own CDs.
Date/Time Properties	System Settings➔Date & Time	redhat-config-date	Set data and time manually or using an NTP server.
User Mount Tool	System Tools➔Disk Management	usermount	Mount and format CDs, floppy disks, or other removable media.
Display Configuration	System Settings➔Display	redhat-config-xfree86	Configure your X desktop.
Domain Name Service	System Settings➔Server Settings➔Domain Name Service	redhat-config-bind	Configure a Domain Name System server.
Apache Configuration	System Settings➔Server Settings➔HTTP Server	redhat-config-httpd	Set up an Apache Web server.
Add new Device Type (Ethernet, ISDN, Modem, xDSL, and so on)	System Tools➔Internet Configuration Wizard	internet-druid	Establish an Internet connection over modem, Ethernet, or other media.

Continued

Table B-1 *(continued)*

Window Name	*From Red Hat Menu, click:*	*From Terminal window, type:*	*Descriptions*
Keyboard	System Settings → Keyboard	redhat-config-keyboard	Choose a keyboard type.
Kickstart Configurator	System Tools → Kickstart	redhat-config-kickstart	Create a Kickstart file you can use to do unattended Red Hat Linux installs.
Language Selection	System Settings → Language	redhat-config-language	Choose a default language (from those that are installed).
GDM Setup	System Settings → Login Screen	gdmsetup	Change the look and feel of the login screen.
Mail Transport Agent Switcher	System Tools → More System Tools → Mail Transport Agent Switcher	redhat-switchmail	Switch between sendmail and postfix mail transports.
Mouse Configuration	System Settings → Mouse	redhat-config-mouse	Change your mouse configuration.
Network Configuration	System Settings → Network	neat	Configure and add network interfaces
NFS Server Configuration	System Settings → Server Settings → NFS Server	redhat-config-nfs	Share directories using NFS
Printer Configuration	System Settings → Printing	printconf-gui	Configure a printer.
Red Hat Network Configuration	System Tools → Red Hat Network	up2date	Register with Red Hat Network.
Root Password	System Settings → Root Password	redhat-config-rootpassword	Get a new root password.
Security Level Configuration	System Settings → Security Level	redhat-config-securitylevel	Set up your firewall.

Window Name	*From Red Hat Menu, click:*	*From Terminal window, type:*	*Descriptions*
Service Configuration	System Settings → Server Settings → Services	redhat-config-services	Change which services run at different run levels.
Audio Devices	System Settings → Soundcard Detection	redhat-config-soundcard	Configure your sound card.
System Logs	System Tools → System Logs	redhat-logviewer	View the contents of system log files.
Red Hat User Manager	System Settings → Users & Groups	redhat-config-users	Add, delete, and change users and groups.

You should have a pretty good feel for how the GNOME desktop and some basic Red Hat tools work now. At various times during the Linux Toys projects, I ask you to run an application to set up the project. In most cases, I either have you launch the application from the Red Hat Menu or run a command in a Terminal window so you can start up an application from the shell. "What is a shell?" you ask.

The shell

The shell is a command line interpreter. To use the shell, you type a command, followed by zero or more options or arguments. An option typically tells the command what mode to operate in while an argument tells the command what item to act on (such as a file or directory). Before you can use a shell, however, you have to get to it.

Getting to a shell

There are several ways you can get to a shell to start entering commands:

- **The shell is all you have.** If you set up Linux to run in text mode, when you log in, the first prompt you see is a shell prompt (no GUI is running). The shell is running and ready for you to start typing commands.
- **A Terminal window.** If you are using a GUI, you can open a Terminal window. That window presents you with the same type of prompt and is ready to take your commands. To open a Terminal window from the GNOME desktop, click the right mouse button anywhere on the desktop and select New Terminal. Figure B-3 shows an example of a Terminal window.
- **A virtual terminal.** Type **Ctrl+Alt+F1** (or press F2, F3, F4, F5, or F6) to open a virtual terminal. You can log in and use the shell from this virtual terminal. Type **exit** to log out when you are done. Then type **Ctrl+Alt+F7** to return to the GNOME desktop.

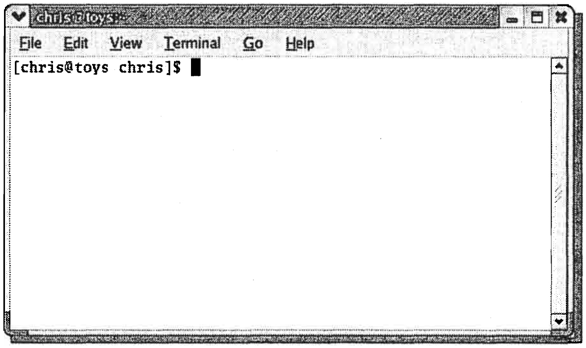


FIGURE B-3: Type shell commands from the desktop using a Terminal window.

Running shell commands

There are hundreds of commands you can run with Linux. But how do you know what commands are available and how they work when all you see after a shell starts is something like this:

```
[joe@toys memos]$
```

To start off, you can tell a few things about your shell session just from the default prompt. In the case just shown, the user name is `joe`, the local computer name is `toys` and the current directory is `memos` (although the full path to the directory is likely something like `/home/joe/memos`). The dollar sign ($) indicates that `joe` is just a regular user, as opposed to a pound sign (#), which would indicate that you are logged in as the root user.

Here is a quick procedure you can follow to familiarize yourself with the shell and your current environment:

1. Type:

   ```
   $ echo $PATH
   /bin:/usr/bin:/usr/local/bin:/usr/bin/X11:
        /usr/X11R6/bin:/home/joe/bin
   ```

   When you type a command, the PATH tells the shell which directories to look in to find the command you want. Most user commands are in `/bin` and `/usr/bin`. There are also some very basic commands (such as `cd`, `history`, and `logout`) built into the shell.

   Commands that you add yourself might be put in `/usr/local/bin` (to be available to all users on your computer) or `/home/joe/bin` (to be just available to joe, or whatever user name you substitute for joe). We put many of the Linux Toys commands in `/usr/local/bin`. Commands in `/usr/bin/X11` and `/usr/X11R6/bin` are those that are related to your desktop and often launch GUI windows.

If you are the root user, you will see /sbin and /usr/sbin in your PATH as well. If, in the course of building a Linux Toy, you see command not found responses, it may be because I am asking you to run a command as the root user and you are logged in as a regular user. Log in as root (or use the su - command to become root), and the commands in /sbin and /usr/sbin will become available to you.

2. Type:

```
$ env | less
HOSTNAME=toys.linuxtoys.net      # Your local computer's name
TERM=xterm                       # Type of Terminal window used
SHELL=/bin/bash                  # The type of shell you are using
HISTSIZE=1000                    # Previous commands being saved
USER=joe                         # Your user name
MAIL=/var/spool/mail/joe         # The location of your mail file
PWD=/home/joe                    # Your present working directory
LANG=en_US.UTF-8                 # The language you are using
HOME=/home/joe                   # Your home directory
LOGNAME=joe                      # Your login name
DISPLAY=:0.0                     # Location of your GUI window
:
```

The env command shows information about your current shell environment. Press the Enter key to page through the output. Then type q when you are done. I edited down the output to show some information that might interest you, then added comments to explain what each item represents.

3. Type:

```
$ man ls
```

This runs the man command, which prints a manual page for the command (in this case the ls command) you enter. You can use the man command to see descriptions of a command, along with what available options. The info command is another useful tool for learning about other commands (for example, type **info ls**).

4. Try some commands for moving around the file system. Type the following to see your home directory name:

```
$ echo $HOME
```

Type these commands to go to your home directory, the /tmp directory, or the /bin directory:

```
$ cd $HOME
$ cd /tmp
$ cd /bin
```

Type this to see your current directory:

```
$ pwd
```

Type these three commands to show a short list of the contents of your home directory, a long list, and a list that also includes hidden files (that begin with a dot):

```
$ ls $HOME
$ ls -l $HOME
$ ls -a $HOME
```

There are lots of options you can use with ls. For example, `-t` sorts files by the time they were last modified, `-C` lists files in columns, and `-S` lists files by size. Often, you will use options together (for example, `ls -lS` lets you sort files by size and show the actual size, along with other information, about each file as well).

5. Try some commands for working with files and directories. Type:

```
$ mkdir $HOME/info
```

This creates a directory named `info` in your home directory. To remove that same directory, type:

```
$ rmdir $HOME/info
```

To create an empty file (in this case, in your home directory named `hat`), use the `touch` command as follows:

```
$ touch $HOME/hat
```

To rename a file or directory (or actually, move it to a different name), use the `mv` command as follows:

```
$ mv $HOME/hat $HOME/hatbox
```

To make a copy of a file, use the `cp` command as follows:

```
$ cp $HOME/hatbox $HOME/hatbox2
```

To remove a file, use the `rm` command as follows:

```
$ rm $HOME/hatbox
```

Using special characters

Some characters have special meanings to the shell. These special characters can save you some typing or tell the shell where to take information from or send it to. I'm going to show you a few useful special characters here.

Use an asterisk (*) to match any combination of characters. For example, type the following:

```
$ ls /usr/bin/redhat-config*
```

The previous command results in a list of all commands in `/usr/bin` that begin with `redhat-config` (these are GUI administration tools that I described earlier). The asterisk can also go at the beginning or in the middle, to match filenames that way (for example, *config-printer or redhat*printer would match the filename redhat-config-printer).

Use a question mark (?) to match any single character. For example, the following would list filenames `hat`, `cat`, and `bat`, but not `that` or `chat`:

```
$ ls ?at
```

Greater than (>) and less than (<) signs are used to redirect input and output to and from commands. Here are a couple of command lines you can try:

```
$ ls /etc > /tmp/etc_list
$ ls /etc/sysconfig >> /tmp/etc_list
```

In the first preceding example, a list of all files in `/etc` are directed (>) to the file `/tmp/etc_list`. (If `/tmp/etc_list` exists, this action erases it to create the new content.) In the second example, a list of files from `/etc/sysconfig` is added (>>) to the `/tmp/etc_list` file. (The double greater-than signs cause the file to be added to, rather than overwritten.)

You can use the less-than sign to direct data to a command. For example, to print the contents of your `/etc/passwd` file, you could type:

```
$ lpr < /etc/passwd
```

While those little arrow characters can pass data between files and commands, the pipe character (|) can pass data from one command to another. For example, to list the contents of the file, send that list through the sort command (to put it in alphabetical order), and print the sorted list, you could type the following:

```
$ cat /tmp/etc_list | sort | lpr
```

Other characters you might want to become familiar with are the backslash (\) and quotes ("). If you want to add a special filename (generally not a good idea, but you can if you like), you can use a backslash or a set of double quotes. In the following two examples, you would create the files abc*123 and abc?&123, respectively:

```
$ touch abc\*123
$ touch "abc?&123"
```

To learn more about special characters, refer to the bash man page (type `man bash`).

Using command history

The shell saves a history of the commands you type (in fact, it saves the 1,000 most recent commands you typed). If you type long path names or complicated command options, there's no reason to retype them. Just recall a command you ran previously, run it again as it is, or edit the command line to do something different.

To see a list of the commands in your history, type:

```
$ history | less
```

Press the space bar to page through your history; then type **q** to quit. To just see the past 10 (or other number) command lines you ran, type the following:

```
$ history 10
```

There are a few different ways you can run a previous command. Using an exclamation mark (`!`), you can refer to command lines from your history by number or by position in the list. Here are a few examples:

```
$ !24
$ !-5
$ !!
```

The first example (`!24`) runs the 24th command line in your history list. The next example (`!-5`) runs the command line you ran five commands ago. The last example (`!!`) repeats the previous command you ran.

To step through your command history, use the up and down arrows. If you land on a command you want to run, but might want to modify, use the right and left arrows to move to the part of the command line you want to change. Then just type to add to the line, use the Backspace key to delete previous characters, or use the Delete key to delete the current character. Press Enter to run the modified command.

Note

You can use a text editor to search and modify commands from your history list. Because I use the `vi` text editor, I always set my command line editor by adding this line to my `$HOME/.bashrc` file:

```
set -o vi
```

With that set, I can use `vi` features to find and edit commands from my history. For example, I press Esc, type `/cat`, and press Enter. The shell finds the previous command I ran that includes `cat` in it. This is very useful if you can remember part of a long command name or path that you want to use in a new command.

Using command completion

Command completion in the shell can save you a lot of keystrokes and mistyped commands. Type part of a command or path name; then press the Tab key. If the command or path is unique (as far as you have typed), the shell will complete the rest of the command or path. For example, type:

```
$ abi<Tab>
```

That's the letters abi, followed by pressing the Tab key. Because I have only one command in my path that starts with the letters `abi`, the shell fills in the rest (in this case, `abiword` appears). Next, you can do the same thing with a filename to go with the command. Let's say there is a file in your current directory named `mybigfile.abw`. If no other file in your current directory begins with `my`, you can type:

```
$ abiword my<Tab>
```

The shell fills in the filename `mybigfile.abw`. If there are more files beginning with `my` in the directory, press the Tab key a second time. A list of files beginning with `my` appears. You can then fill in a few more letters to make your match unique, and then press Tab again to have it complete.

Digging deeper into the shell

Have I told you everything there is to know about the shell yet? Not by a long shot. But I'm running out of room and this is a book about toys, after all. If you find yourself way into Linux from this book, I recommend that you get a more complete understanding of the shell in the following areas:

- **Shell programming** — You can put together groups of commands and functions and run them as shell programs. Most Linux system administrators commonly create shell scripts to do everyday system tasks. The start-up scripts that launch some of the projects in this book are shell scripts.
- **Shell administrative commands** — Although there are graphical interfaces for doing most system administration tasks these days, many times their equivalent text-based commands will be more powerful and flexible. Also, to administer computers over a network, it is often more convenient (or necessary) to run commands instead of GUI tools. Common administrative commands for checking file systems (`fsck`), adding users (`useradd`), and administering printers (`lpadmin`) are often found in the `/sbin` or `/usr/sbin` directories.
- **Shell environment variables and expansions** — I've touched on a couple of environment variables ($HOME and $PATH), but there are dozens more that hold bits of information about your shell environment. There are also ways of expanding arithmetic expressions, filenames, and the output of commands to include in the shell commands you run. Using these items can make it more efficient for you to use the shell.

See the "More Stuff on Linux" section for some pointers to more information on the shell, as well as other aspects of Linux you may want to learn about.

Understanding How Linux Is Organized

Knowing where Linux keeps things on your computer will help you know where to look when something goes wrong. Here is a quick rundown of how files and directories are organized in Linux:

Administrative files

Most basic administrative files are contained in the `/etc` directory and its subdirectories. Most administrative files in `/etc` are plain text files, while a few others are data files created from the output of the text files. While GUI windows are preferred for working with these files (they will lead you through and do some error checking), there are times when there's no other way to fix something than to edit a file in `/etc` directly.

Some important files in `/etc` include passwd (listing all users on the system), hosts (listing host names and IP addresses), printcap (containing information about your printers), and services

(assigning port numbers to network services). Subdirectories of `/etc` that might interest you include:

- **inittab** — Contains system start-up scripts for launching system services. (These scripts are actually linked to files in `rc.d`, `rc1.d`, `rc2.d`, `rc3.d`, `rc4.d`, `rc5.d`, and `rc6.d`, so they can be started and stopped at different run levels.)
- **cron*** — There are cron directories that let you set commands to run at particular times of the day, week, month, or year. These are useful for having such things as system backups or other maintenance tasks run unattended during off-peak hours. Cron directories include `cron.d` (basic cron entries), `cron.daily`, `cron.hourly`, `cron.monthly`, and `cron.weekly` (for commands that run daily, hourly, monthly, or weekly, respectively). In Linux Toys, we use cron entries for getting continual updates on TV listings and temperature data.
- **profile.d** — Contains default configuration settings that are added to a user account when a new user is created
- **skel** — Contains default configuration files that are added to a user account when a new user is created. These files help configure a user's bash shell, GNOME, and KDE desktop. (Type **`ls -a /etc/skel`** to see the files.)
- **sysconfig** — Contains configuration files created by many of the Red Hat configuration tools
- **samba** — Includes files for configuring Windows file and printer sharing (SMB service)
- **X11** — Contains files for configuring your video card and monitor (for the X Window System)
- **xinetd.d** — Contains files used to indicate which network services are available on your Linux system

Server and spool files

The `/var` directories are intended to contain data that you share with the public and log files that contain data about activities on your system. Here are a few subdirectories of `/var` that may interest you.

- **log** — Contains log files of system activity. These include `messages` (general system security messages), `boot` (messages as system services start), `dmesg` (messages as the kernel starts), `lastlog` (messages about login attempts), `maillog` (messages about the mail service), `secure` (security-related messages), `xferlog` (messages about file transfers), and `XFree86` (messages about how your video card and display are working).
- **ftp** — Contains the files and directories you share, if you have set up Linux as an FTP service
- **spool** — Contains spool files (generally, temporary storage) needed by services such as cron (`cron` and `at` directories), printing (`cups` and `lpd`), e-mail (`mail` and `postfix`),

Windows file and printer sharing (samba), proxy service (squid), and Red Hat Network updates (up2date)

- **www** — Contains the Web pages and related content you share, if Linux is configured as a Web server

User files and directories

Users generally keep all the files and directories they create in their home directories, located as subdirectories of /home. It is intended that, if there are multiple people using a Linux system, each person can set up his or her own environment and save private files using these separate /home directories.

Unless you set up directories for shared projects or to allow users a place to add their own Web or FTP content, users aren't expected to change files outside of their /home directory or the /tmp directory. Even Web and FTP content is often shared from subdirectories of users' own home directories. (See Chapters 11 and 12 for examples of using a user's own home directory to add content for Web pages.)

Device files

Files are used to represent nearly everything in Linux. Hardware devices are no exception. Access to your sound card, display, modem, floppy drive, CD drive, hard disk, and other types of hardware is gained through files in the /dev directory.

As a user, you can mostly ignore the /dev directory. However, there are cases where you need to know the names of your hardware devices. For example, you may need to mount a CD (/dev/cdrom) or floppy disk (/mnt/fd0). Or you may need to identify the location of a modem (/dev/modem), video source (/dev/video0), or a particular Terminal window (/dev/pts/*).

Permanent commands and source code

User commands are stored in /bin and /usr/bin. Administrative commands are in /sbin and /usr/sbin.

In the old UNIX days, almost everything was stored under the /usr directory (commands and data). Later, when data that was expected to change was moved to the /var directory structure, /usr was intended to be a more stable directory structure that could be safely shared among multiple machines in read-only mode.

We use subdirectories to /usr (in particular /usr/local) to install many of our Linux Toys projects. Another subdirectory of /usr that may interest you is /usr/src. The /usr/src directory is where the source code for the Linux kernel and other software packages is installed.

Understanding User Logins and Permissions

One of the reasons Linux (and its predecessor UNIX) is considered to be so secure is that it was designed from the start to be a multiuser system. By assigning each user a user name and password, a Linux administrator can easily set up boundaries that define what each person can and cannot do to the computer system.

Every Linux system must have at least one user account, the `root` user. You assign a password for root when you install Red Hat Linux. Keep that password safe, because without it you won't be able to administer your system (and anyone who has your root password will be able to do most anything they like to your system).

Besides the root user account, you should also have at least one regular user account. The name of that account can be anything you like (although typically you would use some version of your real name as your user name). The first time you boot Linux after installation, you will be asked to add a regular user account as part of the firstboot program. (See Appendix C for a description of the firstboot program.)

With separate user accounts assigned, as the administrator of your Linux system you can define permissions related to what each user can and cannot do:

- **Read** — Permissions to read a file allow the user to view its contents.
- **Write** — Permissions to write a file let the user change or delete the file.
- **Execute** — Permissions to execute a file (typically an application) let the user run the file as a program.

The next few sections describe what you should know about the root user and how permissions work in Linux.

Being root . . . the super user

The root user account is referred to as the super user (`su`), because when you are logged in as root you can change any application, file, or directory on the system. Because the root user can be used to cause a lot of damage to your Linux system if you are not careful, I want to give you a few tips about using the root account:

- **Mostly log in as a regular user** — If you are using Linux to browse the Web or play a game, don't log in as the root user. Have a regular user account for doing most everything except system administration. This will help protect you from carelessly damaging your system.
- **Start GUI tasks as a regular user** — If you try to open a window, for example to add a user or configure networking, you can do so as a regular user. The system simply prompts you for the root password before it will let you do the task.

- **Use su to run commands** — If you need to run a few quick commands as the root user, open a Terminal window from your regular GUI session and type the command:

  ```
  $ su -
  Password: *******
  ```

  After you enter the root password, you have full root privileges to run administrative commands. Adding the hyphen (-) after `su` ensures that the root user's environment is run (so that `/sbin` and `/usr/sbin` are added to your available commands and root login scripts are run). After you are done using the root account, type **exit** or **Ctrl+D** to exit the su shell and return to regular user permission.

- **Be careful** — As root user you can really trash your Linux system if you are not careful. Make backup copies of configuration files before you change them. Use asterisks (*) and other special matching characters carefully (`rm -rf *` will remove everything below your current point in the file system).

Caution In particular, when you are working from the shell, anything you change or remove from the files and commands in Linux will be permanently changed. Again, keep backup copies of configuration files and use commands such as `rm` very carefully!

If you are using Linux to build some of the standalone toys in this book, protecting the root password may not be so critical. Add your Linux computer to the network, however, and a poorly secured system can be attacked and used to attack other systems on your LAN or on the Internet. The first step is having a good (and private) root password. Other security measures are described in Chapter 11, "Be a Mini ISP."

Setting file and directory permissions

Each file, directory, application, and device in Linux has permissions associated with it. You can see what those permission are using the `ls` command with the long list (`-l`) option. Here is an example of the `ls -l` command:

```
$ ls -l $HOME/.bashrc
-rw-r--r--    1  joe     tech     124    Mar 13 03:01   /home/joe/.bashrc
```

Permissions for files and directories come in three sets of three settings: Read, Write, and Execute. If all permissions are turned on, they appear as:

```
rwxrwxrwx
```

The first three permission bits define read (r), write (w), and execute (x) permissions for the owner of the file. The next three apply to the group to which the file belongs, and the third set applies to what every other user on the system can do.

In the preceding example, the permissions on the .bashrc file (which holds settings for your shell) is -rw-r--r--, which indicates that the file's owner (which is joe) has permission to read and write this file. Other users in the group tech can only read the file. Likewise, all other users can only read the file.

You can change the permission on any file you own using the chmod command, with either numbers or letters (rwx) as arguments. Read permission is assigned to the number 4, write permission to 2, and execute permission to 1. So, each set of permissions can be set using a number from 0 (no permission) to 7 (full permission). Here is an example:

```
$ chmod 664 $HOME/.bashrc
$ ls -l $HOME/.bashrc
-rw-rw-r--    1  joe   tech   124  Mar 13 03:03 /home/joe/.bashrc
```

In the preceding example, the first 6 sets the owner's permission to read/write, the second 6 sets the group permission to read/write, and the final 4 sets everyone else's permission to just read. To run a command or use a directory, those items must have execute permissions set. For example, type the following:

```
$ ls -ld $HOME
drwx------    1  joe    tech    124    Mar 13 03:03   /home/joe
```

Here you can see that the listing is for a directory (indicated by the d at the beginning of the permissions). The owner (joe) has full permissions to his own home directory (rwx). Group and all other permission are closed, so that nobody can see or access the contents of his home directory (which is the correct thing to do, unless you need to allow access to your directories by, for example, a Web server like the one described in Chapters 11 and 12). If you want to open the contents of your home directory completely (which you shouldn't do), you can type the following:

```
$ chmod 777 $HOME
$ ls -ld $HOME
drwxrwxrwx    1  joe    tech    124    Mar 13 03:03   /home/joe
```

If you went ahead and opened up your directory (even though I told you not to), change it back by typing chmod 700 $HOME. (If the directory is used to hold Web content, use 711 instead, so the Web server can access that content.)

Checking System Processes

If your Linux system is on, there are probably a whole bunch of separate processes running. As with files and directories, each user can control his or her processes, while the root user can control all processes. To see what processes are running, use the ps command as follows:

```
$ ps ax | less
 PID TTY      STAT   TIME  COMMAND
   1 ?        S      0:04  init
     .
     .
     .
1517 ?        S     0:00 syslogd -m 0
1538 ?        S     0:00 /usr/sbin/sshd
1710 ?        S     0:00 sendmail: accepting connections
2399 ?        S     7:09 /usr/bin/ogg123
:
```

The first process you see is `init`, which basically started most of the other ones. After the basic operating system processes, the system services are typically listed. In the preceding example, the `syslogd` daemon is running to log system messages. The `sshd` daemon is running to listen for `ssh` logins from the network. Also, the `sendmail` daemon is waiting to receive e-mail for your system. Lots of other processes will be running for you as well.

In the left column of the `ps` output is the process ID for each process. The `TIME` column shows how much CPU time the process has consumed. The `COMMAND` column shows each entire command line (including options) that was run.

If you are running a GUI, you will notice that `X` consumes a lot of processing power (see the `TIME` column). Watching the `TIME` column is a way to look out for run-away processes. The `PID` column is another column of interest. If you decide you need to kill a process (which you should do only if you're sure you don't need it to be running), you can use the PID to identify it. For example, if the `ogg123` command that precedes didn't stop when you turned off your music jukebox, you could kill it by typing:

```
$ kill -9 2399
```

Or you could use the `killall` command to kill all commands of a certain name. For example:

```
$ killall ogg123
```

Note A way to list all running processes, and then sort them by such things as CPU usage or memory usage, is to run the `top` command. Try running top from a Terminal window or other shell. Then type **M** (sort by memory usage), **P** (sort by percentage of CPU usage), or **T** (sort by CPU time used). Type **q** to exit when you are done.

Using a Text Editor

If you have the GUI running in Red Hat Linux, you can use one of several GUI-based text editors to change configuration files you need to change for Linux Toys. If you don't have a GUI running, you will have to learn to use a text editor from the shell. I'll give you an example of each.

Using gedit (from the GUI)

The gedit text editor comes with the GNOME desktop. It's a fairly basic text editor with a few nice Find and Replace features, along with some easy ways of cutting, copying, and pasting text.

Because most of the files I'll have you edit for Linux Toys are configuration files that are owned by the root user, to use gedit to edit these files you must either:

- Log in as root user to a GUI.

or

- While logged in to a GUI as a regular user, open a Terminal to get root privilege and start `gedit`.

If you are logged in as the root user, open a gedit window from the Red Hat menu by clicking Accessories ➔ Text Editor. You have the permission you need to open any text file at this point, so open the one you want by clicking the Open button and typing (or browsing for) the file you want to edit.

To start gedit so that it can edit files owned by the root user while logged in as a regular user is a bit more difficult. Here's how you can go about it:

1. Right-click on the desktop and click New Terminal. A Terminal window will open.
2. Type **su** - and enter the password for the root user when you are prompted.
3. Type `gedit`, along with the filename you want to open. For example, to open the /etc/hosts file for editing, you could type:

   ```
   # gedit /etc/hosts &
   ```

   I added the ampersand (&) to run the command in the background of the Terminal window. The gedit window should open as shown in Figure B-4.

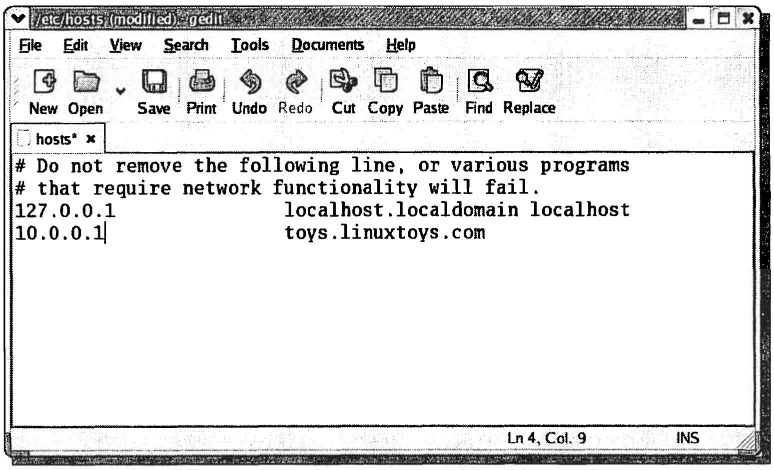


FIGURE B-4: Edit text files from the GUI using the gedit window.

4. Begin editing the file. Click where you want text to go and start typing. Hold and drag the mouse button to highlight text, and then delete it, cut it, or copy it. Move the cursor to a new location and click Paste to paste the text there.
5. Click Save when you are done.

Using vi (from the shell)

Sorry, vi is not as easy to learn as gedit. Once you learn it, however, you never have to lift your fingers off the keyboard to grab a mouse for editing again. As with gedit, you can't change the contents of any file if you don't either own it or have write permission. So if you are editing a configuration file in `/etc`, you should begin as the root user.

To start vi from any shell (virtual terminal or Terminal window), simply type:

```
# vi filename
```

where *filename* is the name of the file you want to edit. The resulting screen that appears is second only to the shell itself as the world's most nonintuitive interface. If you were to edit the same `/etc/hosts` file using vi instead of gedit, the screen would look like this:

```
# that require network functionality will fail.
127.0.0.1      localhost.localdomain    localhost
10.0.0.1       toys.linuxtoys.net
~
~
~
~
~
"/etc/hosts" 4L, 178C                       1,1       All
```

Information at the bottom of the screen shows that the filename is `/etc/hosts`, that there are four lines and 178 characters in the file, that you are currently on line 1 and character 1, and that the entire text in the file is currently being displayed (All). There are two modes that vi operates in:

- **Command mode** — This is the mode that vi begins in. In this mode, you can move around the file, search for text, or enter a command that lets you begin input.
- **Input mode** — After you type a command that lets you input text, you are in input mode. Text that you type will appear where the cursor is after you enter input mode. To leave input mode, press the Esc key.

Here is a quick tutorial on using vi:

1. Type **vi /tmp/testfile.txt**.
2. Press the **i** key (for insert). The editor just switched from command mode to input mode and you can begin adding text immediately before the cursor.
3. Type **Here are a few words that we can begin testing with.**
4. Press the Esc key. You are back in command mode.
5. Press the **H** key to go to the first character of the first line.
6. Press the **w** key a few times to move ahead a word at a time.
7. Press the **b** key a few times to move back a word at a time.
8. Type **o** to open a line below the current line and type another line of text (then Esc to return to command mode).

9. Use the **j, k, l,** and **h** keys to move up and down and across the file one character (or one line) at a time.
10. Put your cursor on the first character of a word and type **dw** to delete the word (no need to Esc after a deletion).
11. Type **A** and begin typing to add text to the end of the current line. Type **I** to add text to the beginning. (These are opposed to lowercase **a** and **i**, which would add text after or before the current character, respectively.) Press Esc to return to command mode.
12. Type **/*word*** (where *word* is replaced by a word in the file) and press Enter. Type **n** to continue to search for other occurrences of the word in this file.
13. Type **:w** to save changes you made to the file. Type **:wq** to save and exit the file. To quit without saving changes, type **:q!** (which causes all unsaved changes to be lost).

There is a lot more to vi than I've just explained. If you get stuck, type **:help** to get further information on vi.

Chuck's Tech Tip

For a good introduction to vim, execute the command `vimtutor` and follow the instructions. (`vim` is actually the underlying command used instead of `vi` these days.) You'll find it helpful to go through it once or twice a week for a few weeks until you are comfortable with it.

More Stuff on Linux

This appendix is intended to be an introduction to Linux in a limited number of pages. All the topics covered here, as well as quite a few more, are described in greater detail in *Red Hat Linux Bible.* Aside from that, the Internet is your best friend when it comes to getting information about Linux. Here are a few Web resources you can draw on to help you with Linux:

- **Google** (`www.google.com`)—The Google search engine seems to be the search tool of choice for Linux users. If you are having trouble getting your hardware to work, try searching for information about your hardware and Linux. For example, to get information about an old laptop, you might search for "Compaq 1620 laptop Linux." There's a good chance that someone else has run into the same problem and may have found a solution.
- **Linux Documentation Project** (`www.tldp.org`)—This site is a tremendous repository for HOWTOs, guides, FAQs, and man pages related to Linux.
- **Red Hat Web site** (`www.redhat.com`)— Red Hat Inc. has excellent support and documentation resources for its products available from its own Web site. If you find yourself making more of your Linux systems than just toys, you may even consider registering your Red Hat Linux system to get technical support from Red Hat.
- **Mailing lists**— Nearly every topic related to Linux has one or more mailing lists available to discuss it. For general information, your local Linux User Group (LUG) may have a mailing list and occasional general meetings where you can get help.

Last, but not least, come and check out `linuxtoys.net`. Although the site is new, we advocate its becoming a gathering place for people who want to keep up with changes and improvements to our Linux Toys projects.

Summary

Linux can be daunting if you are just getting started with it and have no experience with UNIX-type computer systems. This appendix attempts to break the ice a bit. If you can learn something here about the Red Hat Linux GUI, the shell, system files, user accounts, and basic text editing, you should be able to get started with the projects in this book. Books such as *Red Hat Linux Bible* and various resources on the Web can help you get a deeper understanding of Linux.

appendix C

Basics of Red Hat Linux Installation

in this appendix

- ☑ Getting a compatible PC
- ☑ Starting Red Hat Linux installation
- ☑ Learning about disk partitioning
- ☑ Booting Red Hat Linux
- ☑ Adding software packages later

The hardest part about installing Red Hat Linux is making sure that you have computer hardware that is compatible with Red Hat Linux. However, if you meet the minimum hardware requirements (CPU, hard disk space, and RAM in particular), a tremendous amount of (even older) PC hardware will work just fine for installing Red Hat Linux.

Take a few minutes to look through this appendix to find what you need to install Red Hat Linux and the procedure you can use to do the actual installation. After installation, check out the review of the Red Hat Linux boot process to learn about how Red Hat Linux gets up and running.

If after installing Red Hat Linux you find that you need to go back and add some software packages, you have a few choices. You can use the Red Hat Add and Remove Software window or the `rpm` command. Both of those tools are described at the end of this appendix.

Wait, You Need Some Stuff

Every Linux Toys project starts with Red Hat Linux on a PC (Pentium class or better). So our Toy requirements begin with Red Hat Linux requirements. Here is what you need before you begin installing Red Hat Linux as the foundation for your Linux Toys.

Note

The requirements shown here are for Red Hat Linux installation. Before you install Red Hat Linux, you need to check on hardware and disk-space requirements for the individual projects you are building as well. Some projects, particularly those that involve putting music or video on your computer, can demand substantially more hard-disk space and may work poorly on slower processors.

Red Hat Linux CDs

Red Hat Linux comes on three installation CDs and three source-code CDs. All you need are the installation CDs (which we assume you have before you begin the Red Hat Linux installation described here). If you don't have them yet, you can get them from:

- ***Red Hat Linux Bible*** — Red Hat Linux comes packaged with several different books. I recommend this one because, well, I wrote it. It also has the complete 3-CD installation set, instead of just the Publisher's Edition (on only one or two CDs). Besides including the CDs, *Red Hat Linux Bible* will fill in a lot of the details on installation and setup that we just don't have the room for in this book. Software from *Red Hat Linux 9 Bible* was used to build and test Linux Toys.
- **Downloads from the Internet** — Go to `www.redhat.com/mirrors` for a list of sites where you can download Red Hat Linux ISO images. If you have at least a DSL or cable modem Internet connection, you can download these images in a matter of hours. Then you need a CD burner to burn each ISO image to CD. There are other ways of installing these images (over the network or from a spare hard disk), but downloading and burning is the most common method.

We based these procedures on Red Hat Linux 9 and the following release. Although the projects will work on other versions of Red Hat Linux, we haven't tested all of them, so you're on your own.

A PC

The personal computer you use for these projects must meet certain minimal requirements. Even then, there are some hardware components that won't work (usually because of some incompatibility imposed by the manufacturer). Here are the minimum requirements for a PC to run Red Hat Linux:

- **Processor** — The CPU must be at least a Pentium-class PC (386 and 486 processors are not supported by Red Hat Linux installation). The CPU must be at least a 200 Mhz processor for text mode or a 400 Mhz processor for GUI mode. (Those numbers relate to the speed at which the processor communicates with other hardware on the computer's bus.)

  Besides actual Pentium CPUs, Red Hat Linux supports compatible processors, such as the AMD K5, K6, K6-2, K6-3, and Athlon processors.
- **Hard-disk size** — Different types of installs require different amounts of hard-disk space. Here are the minimum amounts of disk space you need for each of the types of installs:
  - Personal Desktop — 1.7GB
  - Workstation — 2.1GB
  - Server — 850MB
  - Custom — 475MB (if you select minimum), 5GB (if you select Everything), or somewhere in between

Each Linux Toys project adds its own requirements to Red Hat's requirements. So make sure you check each project you want to build for its disk-space requirements before starting to install Red Hat Linux.

Note

Video and audio projects require a lot of extra disk space to work effectively. Some projects take little extra space. Check disk space needed for each project before beginning.

- **Random Access Memory** (RAM) — The minimum amounts of RAM you need on your PC are 64MB (for text mode) or 128MB (for graphical mode). The recommended minimum for getting decent performance in GUI mode is 192MB. More RAM will improve performance.
- **CD drive** — I'm assuming your computer has a bootable CD drive. If it doesn't, you can start the installation from a floppy disk and install over the network.
- **Hardware components** — CPU, hard disk, and RAM requirements are the basic requirements for Red Hat Linux hardware. Various projects will require other hardware, such as an X-compatible video card, sound card, TV card, modem, or other hardware. The hardware requirements for those items are described in the requirements for each project.

Besides these general rules, you can find out about specific hardware that is known to work or not work. The official way to see if the PC you have is supported is to check out the Red Hat Linux Hardware Compatibility site: `http://hardware.redhat.com/hcl`. If you are using a name-brand PC, I would suggest searching the list of manufacturers to see if your particular model has been tested.

Firing Up Red Hat Linux Installation

Although I'm going to try to give you the quick version of Red Hat Linux installation, there are some things that can go wrong along the way. Most problems have to do with incompatible hardware. For other problems and issues that you might run into, it's not a bad idea to check out the Official Red Hat Linux X86 Installation Guide for some extra tips. You can find it at `www.redhat.com/docs/manuals/linux`. You can select the Installation Guide in HTML or PDF formats. Or you can download the guide in a software package (RPM or tar formats).

Another reason you may want to check out this guide is that there are a lot of different ways to go about installing Red Hat Linux. Right here, we're assuming that you are:

- **Installing from the three Red Hat Linux CDs.** There are also ways of installing from an HTML server, FTP server, NFS server, or hard disk. We don't cover those here.
- **Installing only Red Hat Linux on the PC.** You can install multiple operating systems on a hard disk, provided you have enough disk space. The hard part of that is usually resizing existing partitions to give space to Linux. Older FAT and VFAT partitions (used on Windows 95 and 98) can be resized using the FIPS utility. To read about how to use FIPS, you can insert the first Red Hat Linux installation CD into your running

Windows system and open the file `dosutils/fipsdocs/fips.doc` on the CD. To resize partitions for later Windows systems, you need to purchase a third-party product, such as Partition Magic (`www.partitionmagic.com`).

After that, you just need to understand a bit about disk partitioning. You are certainly welcome to have multiple operating systems on the disk. However, we're describing how to make toys out of PCs, so we're going to describe how to have Red Hat take over the whole PC.

Okay, if you have all your pieces assembled, here's how to install Red Hat Linux.

1. **Insert CD:** Put Red Hat Linux installation disk #1 into your PC's drive and reboot.
2. **Boot screen:** From the boot screen, press Enter. This starts Red Hat Linux in graphical mode. If the screen is garbled, run the install in text mode instead by typing `linux text` at this first prompt.
3. **Media check:** If you are asked to check your media, you may select OK to check that each installation disk is not damaged or corrupted. Or you can select Skip to go ahead with the installation process.
4. **Welcome:** Select Next from the Welcome screen.
5. **Language:** Select the Language to use during installation and click Next. (You can add other languages later.)
6. **Keyboard:** Select the keyboard type you are using and click Next. (The keyboard is typically a combination of languages and an arrangement of keys used. Many languages offer only one keyboard arrangement.)
7. **Mouse:** Select the type of mouse you are using and click Next. (A generic mouse will work in most cases. Choose the mouse by its type and by where it is connected.)
8. **Monitor:** The installation process tries to detect your monitor. If the monitor is detected, click Next to continue. Otherwise, select your monitor (by manufacturer name) from the scrolling list. If your monitor doesn't appear on the list, consult the documentation that comes with your monitor and manually enter the horizontal and vertical sync rates into the appropriate box on the screen. Click Next to continue.
9. **Upgrade?:** If there is a previous version of Linux found on your hard disk, you are asked if you want to upgrade it. You can select to upgrade (which will keep your data intact and upgrade your operating system) or select not to (and erase the contents of the disk to do the new install).
10. **Install Type:** Select the type of install you want. (Each project has a recommended type of install and, often, a minimum type of install.) Here are your options:
    - **Personal Desktop** — Includes a graphical desktop environment and applications that are appropriate for using your computer for running applications at home or at work
    - **Workstation** — Similar to the Personal Desktop but also includes software development and system administration tools

- **Server** — Installs software that lets you configure Red Hat Linux as a file, Web, print, or other type of server. You can choose whether or not to add a graphical interface.
- **Custom** — Lets you customize your Red Hat Linux installation by selecting which packages you do and don't want installed. The Custom install type lets you also select Everything or Minimum install types.

Figure C-1 shows an example of the Installation Type screen.

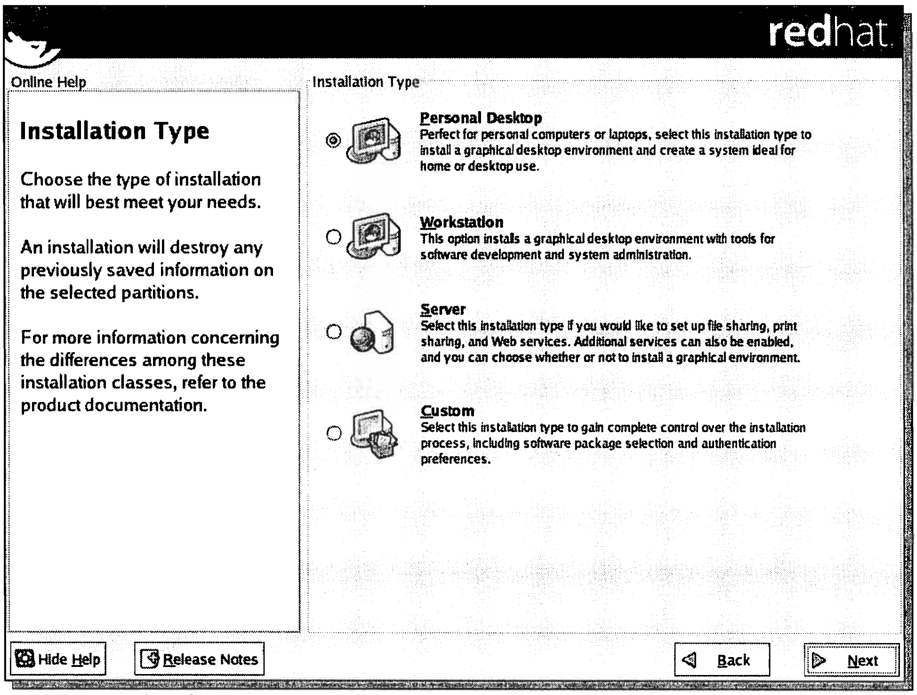


FIGURE C-1: Select from preset installation types or customize your installation.

11. **Partitions:** Select whether you want to have Red Hat Linux automatically partition your hard disk or whether you want to partition it yourself using Disk Druid. You are probably safe to just have Red Hat partition your disk, especially if you are erasing the whole disk. For more information on disk partitioning, see the sidebar "A Lesson in . . . Disk Partitioning."

Figure C-2 shows an example of the Disk Druid screen.

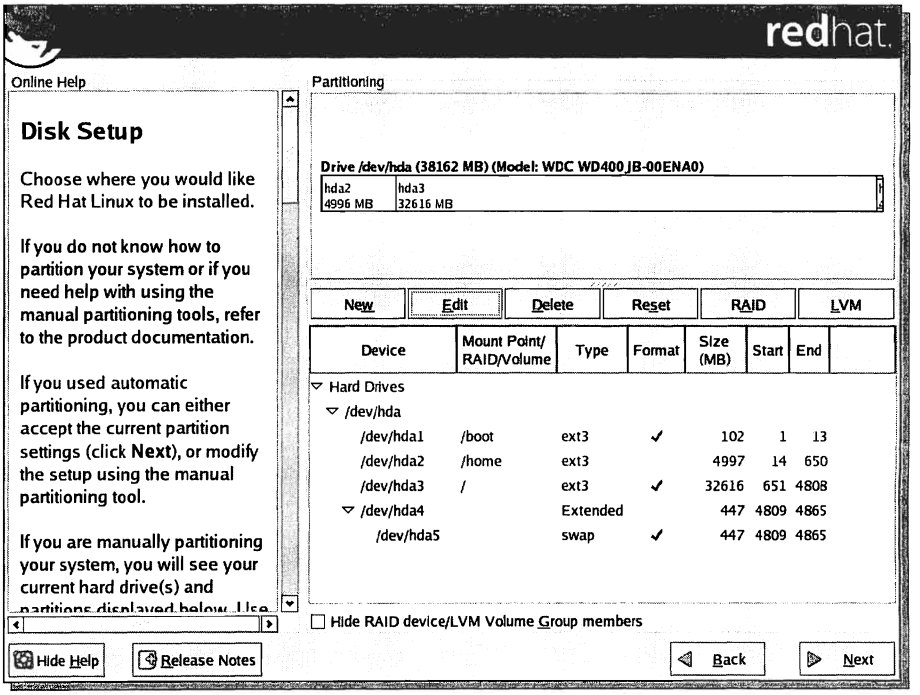


FIGURE C-2: Set disk partitions with Disk Druid.

12. **Erase Partitions?:** You are asked if you want to have disk partitioning done automatically or manually. If you select Automatically, you are asked if you want to remove only the Linux partitions, remove all partitions, or use only free space. (To have the Linux Toys projects take over the whole computer, you can select to remove all partitions. If you want to leave Windows on your computer, remove only the Linux partitions (provided you have enough free disk space to devote to Linux).

    At this point, I also recommend selecting the box marked Review (and modify if needed) the partitions created. This allows you to see what partitions are being created.

13. **Partitions OK?:** If the partitions that are created look good, click Next to continue. If you selected to set partitions manually, you need to assign each partition separately.

A Lesson in . . . Disk Partitioning

Partitioning is a method of splitting up a hard disk into separate, self-contained parts. There are several reasons for dividing a hard disk into separate partitions.

- **Lots of operating systems**—If you have more than one operating system, each must have at least its own bootable partition.
- **Lots of file system types**—To have a Windows NTFS partition and a Linux ext3 file system on the same hard disk, they must be on separate partitions. There are many different file system types, each with its own attributes and uses.
- **Many hard disks**—Each hard disk on your computer must have at least one partition.
- **Setting boundaries**—By having separate file systems on the same operating system, you can protect yourself from running out of disk space on the whole disk at once. For example, if a user fills up the entire partition devoted to his home directory (**`/home`**), applications that are writing to temporary files (in **`/tmp`**) and spool files (in **`/var`**) can keep running.

Linux itself requires that you have at least one swap partition and one Linux partition (usually an ext3 or ext2 file system type). If you let Linux partition your disk, it will create a `/boot` partition as well. The small boot partition makes it quicker to check, mount, and load the bootable kernel than it would if it were stored on a much larger file system.

For Linux Toys, there are several reasons why you might want to go beyond the three partitions (`/`, `/boot`, and swap) that Red Hat assigns automatically. Here are a few ideas:

- **Linux Toys Jukebox**— Installing your whole CD collection on hard disk is a big investment in time. By having a large partition or separate hard disk assigned to `/usr/local` or `/usr/local/share` you can keep your music and CD database separate from the operating system. Later, if you want to reinstall Linux, you can erase everything except your music partition and not risk damaging it.
- **Be a Mini ISP or Web Hosting Service**—If you are sharing a computer with multiple users, you may want to restrict how much disk space each person can use. Besides using features like disk quotas, you should consider separate partitions for **`/home`** (space for user accounts) and **`/var`** (space for Web and FTP server content).

With hard disks coming down in price as much as they have, many administrators get around the disk-space issue by simply having a hard disk that is so large; there is never an immediate danger of running out of disk space. Since we are mostly building toys here, for most of the projects you can get by without fiddling with partitions.

14. **Bootloader:** Click Next to accept the defaults relating to the boot loader. The defaults are:

   - **GRUB** — The Grand Unified Bootloader (GRUB) is installed as your boot loader. Your other choices are to use LILO or no boot loader.
   - **Bootable operating systems** — A list of bootable operating systems are shown in the box. If only Red Hat Linux is installed on this computer, you will probably see only one bootable operating system. If there is more than one, you can select which one is booted by default (see which one is checked).
   - **No boot loader password** — No password is assigned to the bootloader by default. Click the Use a bootloader password box; then select Change password to choose a password for the bootloader. (If you add a password, you will have to enter it before your computer will boot.)
   - **Advanced options** — Click Configure advanced bootloader options if you want to add options to pass to the operating system at boot time.

Chuck's Tech Tip

GRUB doesn't work if you are using mirrored root disks. You need to choose LILO if you want to mirror your root disk by using software RAID.

15. **Network configuration:** If Red Hat Linux detects a network card on your computer, you will be asked to configure it. If you don't know what kind of network you have configured, you can skip this section for now.

Cross-Reference

See Chapter 7 for information on setting your network connections in Red Hat Linux.

16. **Firewall:** If you are asked to configure a firewall, you can choose one of these security levels and click Next to continue:

   - **High** — This is very secure (if you are just using the computer for Web browsing and FTP). Only a few types of connections are allowed at this level to let your computer get addresses and use the Web (not offer any services to the Web).
   - **Medium** — This is fairly secure. It still blocks ports used to offer common services to the Web. You can use this level and then add only those services you want to offer. For example, you can click particular services on this screen that you know you want to offer: DHCP, SSH, Telnet, WWW, Mail, and FTP.
   - **No firewall** — This is insecure and could allow access to services on your computer over the Internet that you have not properly secured. Don't use this level if you are connected to the Internet, unless you are already behind a firewall. If you are connected to a secure LAN, you can choose No firewall.

   Figure C-3 shows an example of the Firewall Configuration screen.

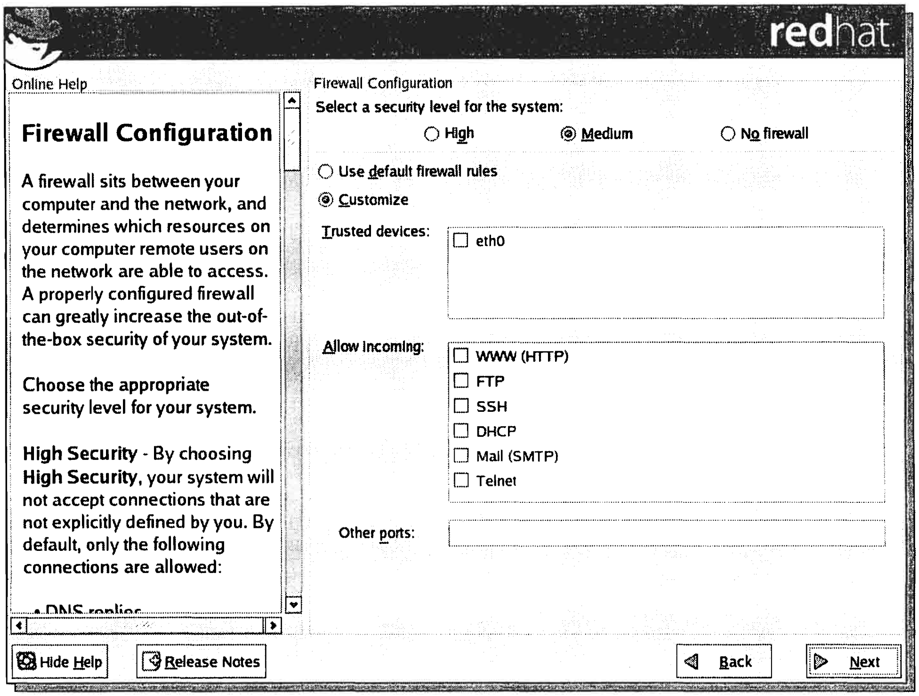


FIGURE C-3: Secure your firewall by using the Firewall screen.

Check if the Linux Toys project you are building requires that a particular service (or port number) be open for the project to work. Again, check out Chapter 7 for information on setting up a firewall to protect the boundary between your computer and the network.

17. **More languages:** Red Hat Linux supports multiple languages on the same system. Choose the language to use by default (it's probably already marked in the box above). Then select any additional languages you want supported on your computer. Click Next to continue.
18. **Time:** You can set your system time based on your location (Location tab) or by the number of time zones from UTC time (UTC offset tab). Location can be selected by clicking on the map or by scrolling to a city in your time zone. Click Next to continue.
19. **Password:** Enter a password (twice) to assign a password to the root user and click Next.

Caution

The root user is what you use to administer the Red Hat Linux system. It is critical that you remember the root user password!

20. **Accept Packages?:** You can either accept the current package list (the screen shows the major categories of software you are installing) or select to customize your software packages. I usually select Custom, so I can add any extra packages at this point that I need for the particular Linux Toys project. (You can always add them after Red Hat Linux is up and running, if you prefer.)
21. **Choose Packages:** If you do decide to choose your own packages, a Package Group Selection window appears. Use this window to select package categories you want, and then select Details to choose individual software packages that you may want to add. Click Next when you have selected the packages you want.
22. **Go for It!:** Nothing has been written to your hard disk yet, so you can still back out now if you want. (To quit, you can remove the CD and press Ctrl+Alt+Del.) But, if you are ready, click Next to begin reformatting your hard disk and installing the software.

    You can watch the progress on the screen and see how much time is left to complete the installation. Insert the #2 and #3 CDs as you are prompted for them.
23. **Boot disk:** After the packages are installed, you are asked if you want to create a boot disk. If your computer has a floppy disk drive, I recommend that you insert a formatted floppy disk, select Yes, and click Next to create the boot disk. Otherwise, click No and Next to continue.
24. **That's all folks!:** If the Congratulations screen appears, you are done. Click Next to end installation, pop out the CD, and your computer should restart automatically.

If everything went well (let's cross our fingers), your computer should reboot. The next section tells you a bit about what Red Hat Linux does during the normal boot process.

Booting for the First Time

When a PC boots up, it starts in the same way (no matter what operating system you are using) from the computer's BIOS, then takes its own direction depending on the type of boot-loader and operating system you are using. For Linux Toys, the boot process is important because, for some projects, we want to have the PC boot to a single-use device (which is done mostly at the end of the process).

After you have installed Red Hat Linux, here are the basic steps your computer goes through when it starts up:

```
BIOS→GRUB→Red Hat Linux kernel→Run level 5→Firstboot→login
```

If you selected all the defaults when you installed Red Hat Linux, your computer goes through the steps just shown when it starts up. Here are some quick descriptions of each of those steps:

BIOS

The Basic Input/Output System (BIOS) is stored in the computer's memory. BIOS is the first thing that the CPU looks for when you power up or reboot the computer. The BIOS program does the most basic checks and tests of the computer hardware to make sure that it is ready to go to the next step.

Once the BIOS identifies the basic hardware, it looks for an operating system to boot. A common order in which it will check for an operating system is:

- Floppy disk
- CD
- Hard disk

The search order for a bootable operating system may be different on your computer. The first two media (floppy and CD) are typically there so that you can override what would normally boot from the hard disk.

Note

You can usually change the boot order from the system setup screen. When BIOS first starts up, it will usually tell you what key to press to enter setup mode. You might want to check that CD comes before the hard disk in the boot order if you are not able to boot from CD. If your computer won't boot from CD, you may need to create an installation boot floppy.

If no bootable CDs or floppy disks are in your computer, the BIOS looks for your hard disk (typically the master IDE device on the first IDE bus). Then the BIOS runs the program that it finds in the Master Boot Record (MBR), which is on the first sector of that device.

In the case of the computer where you have just installed Red Hat Linux, by default the GRUB bootloader will be what the BIOS runs from the MBR.

GRUB

GRUB is the default bootloader for Red Hat Linux. When GRUB starts, it displays the bootable operating systems that were defined when you installed Red Hat Linux. If Red Hat Linux is the only operating system on the computer, you will probably see only one entry on the GRUB screen. If it is a dual boot system with Windows, you will probably see one line for booting Windows and one for booting Red Hat.

At this point you can:

- Wait 10 seconds and the default operating system boots.
- Press Enter to boot the operating system indicated by the cursor.
- Move the arrow keys up and down to select the operating system you want; then press Enter.

GRUB has some editing capabilities (select the boot entry that you are interested in and press **e**). If there is some option you want to pass to the Linux kernel, you can do it by editing the command line. Normally, you shouldn't have to change the boot entries at all. You might want to add to these boot lines if you need to identify some hardware that the kernel isn't detecting or if you want to start at a boot level other than the default.

To permanently change boot entries, or to add new boot entries for additional operating systems or special kernels, you can edit the `/boot/grub/grub.conf` file. For more information on GRUB, see the GRUB man page (type **man grub** from a shell).

Assuming that you are booting Red Hat Linux, the system loads what is called the initial RAM disk (initrd). The initrd makes sure that drivers needed during the boot process are available to the Linux kernel.

Kernel

The kernel contains the most critical parts of the Linux operating system. When it starts up, the kernel initializes hardware and allocates resources.

With the basic Linux kernel up and running, the kernel starts the init process, which starts everything else.

Init

The init process (`/sbin/init`) is responsible for starting up all the basic system services on your Linux system. It runs the programs your system needs based on information in the `/etc/inittab` file. The actions that init takes include:

- **Determining the default run level.** The system will usually start in run level 3 (multiuser mode with a text-based login) or level 5 (multiuser mode with GUI-based login).
- **Running the system initialization script** (`/etc/rc.d/rc.sysinit`). This script does a bunch of stuff, such as setting the computer's host name, setting the system clock, checking and mounting file systems, and loading modules.
- **Starting and stopping run-level scripts.** The init process checks the directory assigned to the run level and starts all processes beginning with "S" in that directory and stops all processes beginning with "K."

The run-level scripts are used primarily to start services that run all the time on your system. This includes features needed for print servers, file servers, Web servers, firewalls, and other types of services.

When we have you use a run-level script with Linux Toys, that run-level script will usually be set up to run automatically from the default run-level directory. That's if you want the service to be on all the time. A lesson in run-level scripts is contained in Chapter 3.

Login

You know that your computer is up and running when you see a login screen (if you boot to run level 5) or a simple text-based prompt. Type the user name and password to log in. The first time you log in, you will have only a root user account and password. Use that name and password to log in and you can add other user accounts later if you need them.

If you log in to a text-based prompt, you will see a shell prompt (either `$` or `#`). If you logged in to a GUI login screen, a GUI desktop appears.

The first time you boot Red Hat Linux, instead of a login prompt, you may see the Red Hat Update Agent screen (also referred to as the firstboot screen).

Right after installation . . . firstboot

The Red Hat Update Agent runs only the first time after installing or upgrading Red Hat Linux. It lets you do some basic configuration.

- **User account** — Add a regular user account for yourself to use for nonadministrative purposes.
- **Date & time** — Set the system date and time. If you have a connection to the Internet, you can get your time synchronized automatically from a network time server.
- **Sound card** — Configure your sound card.
- **Red Hat Network** — Configure your computer to get automatic updates of Red Hat Linux software.
- **Additional CDs** — Install software from additional CDs.

At this point, Red Hat Linux should be up and running and ready to start building your Linux Toys. If, during the course of building the Linux Toys, you find that you need to install additional software packages, you can use either the Red Hat Add and Remove Software window or the `rpm` command. You can find out about these tools in the next sections.

I Need to Install More Packages!

If you try one of the Linux Toys projects and you get a message like "command not found," it may be because the software package that contains that command isn't installed. If you have the X Window System and either GNOME or KDE installed, you can run the Red Hat Add and Remove Software window to install packages. If you have only a shell command line available, you can use the `rpm` command.

Installing packages from the Add and Remove Software window

To launch the Red Hat Add and Remove Software window (while you are logged in to a GUI), click the Red Hat icon on the desktop. From the menu that appears, click System Settings → Add/Remove Applications. Figure C-4 shows the Red Hat Add and Remove Software window.

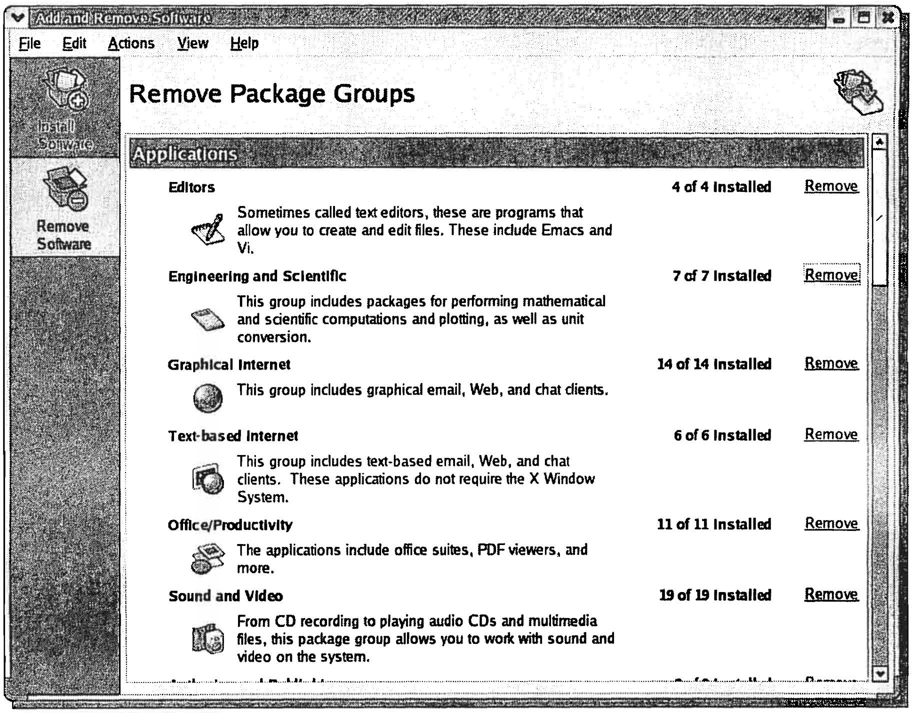


FIGURE C-4: If a Linux Toy needs more Red Hat software, use the Add and Remove Software window to install it.

This window is made to install (or uninstall) software that is part of the Red Hat Linux distribution. (If you need to install an RPM package that is not part of Red Hat Linux, you have to use the `rpm` command.) To use the Add and Remove Software window, here's what you do:

1. Click the Install Software icon in the left column.
2. Scroll through the groups of packages using the scroll bar.
3. Click the Add link next to the group that interests you to see a list of all packages in the group. The numbers next to the link show how many packages are ready to be installed from that group, followed by the total number of packages in the group.

4. Click the package name you want to install (so the checkmark comes on).
5. Click Close to close the details window.
6. Repeat the selection of packages until the list of packages appears as you would like.
7. Click Actions ➔ Install Package. You will be prompted to insert the necessary Red Hat Linux CDs.
8. Close the window after all packages are done installing.

You can go right ahead now and use the software you just installed. Believe it or not, you don't have to reboot. For a new service to take effect, however, you may need to restart that service.

Installing packages with rpm

The command line method of installing RPM packages is the `rpm` command. This command is just loaded with features. You can use it to add, delete, or update any RPM software package. You can also use it to get information about RPM packages (such as to check whether a package is installed, view descriptions of a package, or find out what package a particular file is from).

You can install an RPM package with the `-i` option (install). However, it is more common to use the `-U` option (upgrade), since that command line will work whether the package is currently installed or not. You can begin with the RPM in a directory on your system or by mounting a Red Hat Linux CD and installing from there. Here's an example of how to do the latter.

1. Insert into your CD drive the CD containing the package you want to install.
2. Log in as the root user and open a Terminal window (right-click on the desktop and click New Terminal).
3. If the CD doesn't mount automatically, mount it by typing the following:

   ```
   # mount /mnt/cdrom
   ```

4. Go to the RPMS directory on the CD:

   ```
   # cd /mnt/cdrom/RedHat/RPMS
   ```

Note

Most software on the Linux Toys CD is also in RPM format. With the Linux Toys CD mounted, go to the `/mnt/cdrom/` directory and look for a chapter directory to install Toys software for a particular project.

5. Type the following `rpm` command (with options), followed by the name of the package to install. For example, to install the cdlabelgen package, you could type the following:

   ```
   # rpm -Uhv cdlabelgen*rpm
   ```

   In the preceding example, typing the asterisk (*) just saves me from having to type all the letters and numbers associated with the version of the package. You could also type the first part of the package name and press the tab key to have the shell try to fill in the rest of the package name.

To check that the RPM has been installed, you can use the `-q` option. For example, to see whether the cdlabelgen package is installed, type the following:

```
# rpm -q cdlabelgen
cdlabelgen-2.6.0-1
```

To remove a package that has been installed, you can use the `-e` (erase) option. For example, to remove the cdlabelgen package, you could type:

```
# rpm -e cdlabelgen
```

To remove a package, there is no need to use the entire version number. You have to type only the base name of the package.

If you need more information on the `rpm` command, type `man rpm` from a Terminal window.

Summary

Because Red Hat Linux forms the foundation for the Linux Toys projects in this book, this appendix was created to help you through the process of installing Red Hat Linux. It not only steps you through the installation process; it describes the process of booting your Red Hat Linux system for the first time. If you want to add or remove software packages after Red Hat Linux is installed, you can use the Add and Remove Software window or the `rpm` command.

Index

Continued

D

E

F

R

S

Operating Systems/UNIX & Linux

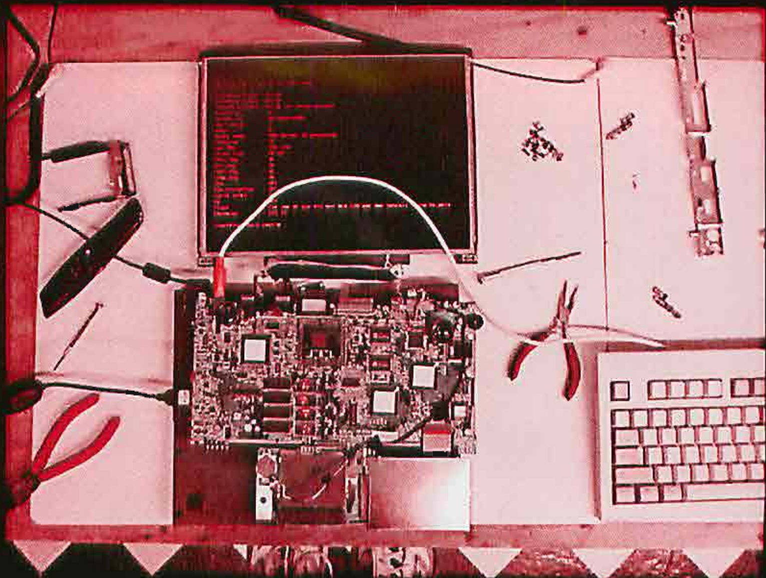
$29.99 US/$44.99 CAN/£27.95 UK

Chris and Chuck welcome you to the Linux toyshop

If the idea of remodeling an old laptop into a digital picture frame has your palms tingling, take this book home. That's one of thirteen cool projects these two experts will teach you to build using a PC, a few spare parts, and a little Linux. You can build single-use projects on an old PC, or simply add "Toys" projects to your high-end Red Hat® Linux® box. Just be sure someone knows where you are — in case there's an emergency before you're finished.

Christopher Negus is the author of all editions of the bestselling *Red Hat Linux Bible* as well as several other computer books. A Linux aficionado, Chris recently wired his house with coax and Cat 5e wiring so he could build more toys.

Chuck Wolber is an experienced Linux system administrator, programmer, and founder of Quantum Linux Laboratories. He's president of the Tacoma Linux User Group and a devout hardware tinkerer.



The Toys

Each with a complete materials list and detailed, illustrated instructions

1. Music jukebox
2. Home video archive
3. Television recorder/player
4. Arcade game player
5. Home network server
6. Home broadcast center
7. Temperature monitor
8. Telephone answering center
9. Mini ISP
10. Web-hosting service
11. DogHouse Linux with BSD games
12. Toy car controller
13. Digital picture frame

CD-ROM INCLUDES

- Xmame classic console gaming player
- WebVCRplus Web-based TV recording software
- VOCP telephone answering and voice messaging software
- Ffmpeg for recording and streaming audio/video
- BSD-Games classic keyboard games package
- Mplayer video player for Linux

System requirements: PC compatible with Red Hat Linux. Linux Toys CD RPM packages were tested to run on Red Hat Linux 9. Linux Toys source code provided to build toys on other Linux systems. Readers must supply their own copies of Red Hat Linux 9.

7 85555 86224 1

WILEY
wiley.com

Companion Web site

Find additional information about Linux Toys at www.wiley.com/compbooks/negus

ISBN 0-7645-2508-5

52999

9 780764 525087